SIDDHARTH INSTITUTE OF ENGINEERING &TECHNOLOGY :: PUTTUR

(AUTONOMOUS)

Masters in Computer Applications (MCA)

Course Structure

MCA I Year - I Semester

S. No.	Course Code	Subject	L	T	P	C
1	25MC9101	Mathematical Foundation of Computer Science	4	-	-	4
2	25MC9102	Computer Organization & Architecture	4	-	-	4
3	25MC9103	Software Engineering	4	-	-	4
4	25MC9104	Data Structures Using C	4	-	-	4
5	25MC9105	Database Management Systems	4	-	-	4
6	25MC9106	Software Engineering Laboratory	-	1	2	2
7	25MC9107	Data Structures using C Laboratory	-	1	2	2
8	25MC9108	Database Management Systems Laboratory	-	1	2	2
9	25MC9109	Research Methodology and IPR	2	-	-	2
		Contact periods/Week	22	3	6	28
			Total/Week 31			

MCA I Year - II Semester

S. No.	Course Code	Subject	L	T	P	C
1	25MC9110	Operating Systems	4	-	-	4
2	25MC9111	Data Mining and Business Intelligence	4	-	-	4
3	25MC9112	Python Programming	4	-	-	4
4	25MC9113 25MC9114 25MC9115	**Programme Elective – I** Computer Networks Software Testing Methodologies Managerial Economics and Financial Accountancy	3	-	-	3
5	25MC9116 25MC9117 25MC9118	**Open Elective – I** Operations Research Digital Marketing Cloud Computing	3	-	-	3
6	25MC9119	Operating Systems Laboratory	-	1	2	2
7	25MC9120	Data Mining and Business Intelligence Laboratory	-	1	2	2
8	25MC9121	Python Programming Laboratory	-	1	2	2
9	25MC9122	**Skill Oriented Course – I** JAVA Programming	1	-	2	2
10	25MC9123	**Seminar**	-	-	4	2
		Contact periods/Week	19	3	12	28
			Total/Week 34			

MCA II Year -I- Semester

S. No.	Course Code	Subject	L	T	P	C
1	25MC9124	Web Technologies	4	-	-	4
2	25MC9125	Big Data Technologies	4	-	-	4
3	25MC9126	DevOps & Agile Programming	4	-	-	4
4	25MC9127 25MC9128 25MC9129	**Programme Elective – II** Software Architecture & Design Patterns Network Security Machine Learning	3	-	-	3
5	25MC9130 25MC9131 25MC9132	**Programme Elective – III** Mobile Application Development Internet of Things Block Chain Technologies	3	-	-	3
6	25MC9133	Web Technologies Laboratory	-	1	2	2
7	25MC9134	Big Data Technologies Laboratory	-	1	2	2
8	25MC9135	DevOps & Agile Programming Laboratory	-	1	2	2
9	25MC9136	Summer Internship / Industry Oriented Mini Project/ Skill Development Course (Minimum 6 weeks)	-	-	-	2
10	25MC9137	**Skill Oriented Course – II** MEAN Stack Development	1	-	2	2
Contact periods/Week			19	3	8	28
			Total/Week 30			

MCA II Year-II- Semester

S. No.	Course Code	Subject	L	T	P	C
2	25MC9138 25MC9139 25MC9140	**Programme Elective – IV** Deep Learning Social Media Analysis Multimedia Systems and Tools	3	-	-	3
3	25MC9141 25MC9142 25MC9143	**Open Elective – II** Cyber Laws Entrepreneurship NOSQL Databases	3	-	-	3
4	25MC9144	Project Work	-	-	20	10
5	25MC9145	Comprehensive Viva Voce	-	-	-	2
Contact periods/Week			06	00	20	18
			Total/Week 26			

Note: L: Lecture Hour, T: Tutorial, P: Practical, C: Credit
Total Credits: 102

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
4	-	-	4

(25MC9101) MATHEMATICAL FOUNDATION OF COMPUTER SCIENCE

COURSE OBJECTIVES

The objectives of this course:

1. *Introduces the elementary discrete mathematics for computer science and engineering.*
2. *Topics include formal logic notation, methods of proof, induction, sets, relations, graph theory, permutations and combinations, counting principles; recurrence relations and generating functions*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Demonstrate propositional logic, predicate logic, and rules of inference to construct and evaluate valid mathematical proofs.*
2. *Apply sets, functions, sequences, relations, principles of discrete probability, and their properties to model and solve problems in computer science.*
3. *Analyze algorithms of growth functions and complexity; utilize mathematical induction and recursion techniques for algorithm design and correctness verification.*
4. *Use Bayes' theorem, expected value, variance, and advanced counting techniques including recurrence relations and generating functions to solve complex combinatorial problems.*
5. *Model and analyze graph structures, apply graph theory concepts such as connectivity, Eulerian and Hamiltonian paths, shortest-path problems, planar graphs, and graph coloring in computer science contexts.*
6. *Develop the ability to apply mathematical foundations to formal reasoning, algorithm analysis, and problem solving in various areas of computer science.*

UNIT-I

The Foundations Logic and Proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.

UNIT-II

Basic Structures, Sets, Functions, Sequences, Sums, Matrices and Relations: Sets, Functions, Sequences & Summations, Cardinality of Sets and Matrices Relations, Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

UNIT-III

Algorithms: Algorithms, the Growth of Functions, Complexity of Algorithms.

Induction and Recursion: Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness

UNIT-IV

Discrete Probability and Advanced Counting Techniques: An Introduction to Discrete Probability, Probability Theory, Bayes' Theorem, Expected Value and Variance.

Advanced Counting Techniques: Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

UNIT-V

Graphs: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.

TEXT BOOKS

1. Discrete Mathematics and Its Applications with Combinatorics and Graph Theory- Kenneth H Rosen, 7thEdition, TMH.

REFERENCES

1. Discrete Mathematical Structures with Applications to Computer Science-J.P. Tremblay and R. Manohar, TMH,
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe L. Mott, Abraham Kandel, Teodore P. Baker, 2nd ed., Pearson Education.
3. Discrete Mathematics- Richard Johnsonbaugh, 7th ed., Pearson Education.
4. Discrete Mathematics with Graph Theory- Edgar G. Goodaire, Michael M. Parmenter.
5. Discrete and Combinatorial Mathematics - an applied introduction: Ralph.P. Grimald, 5th edition, Pearson Education.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
4	-	-	4

(25MC9102) COMPUTER ORGANIZATION & ARCHITECTURE

COURSE OBJECTIVES

1. *Learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design*
2. *Understand the structure and behavior of various functional modules of a computer.*
3. *Discuss the techniques that computers use to communicate with I/O devices*
4. *Study the concepts of pipelining and the way it can speed up processing.*
5. *Describe the basic characteristics of multiprocessors*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the structure of a computer system, functional units, bus architecture, instruction formats and addressing modes of the INTEL-8086 architecture.*
2. *Apply knowledge of computer arithmetic for signed and unsigned numbers, and design efficient hardware units such as fast adders and multipliers.*
3. *Analyze the architecture and functioning of the basic processing unit including execution of instructions, control unit design, and bus organization.*
4. *Develop memory hierarchy concepts, including semiconductor memories, cache organization, virtual memory, and secondary storage to optimize performance.*
5. *Implement I/O organization including interrupt handling, DMA, interface circuits, and standard input/output protocols.*
6. *Use advanced concepts in computer architecture such as pipelining, hazard detection, and parallel processing in multiprocessor systems and interconnection networks.*

UNIT-I

Basic Structure of Computer: Computer Types, Functional Units, Basic operational Concepts, Bus Structure, Software, Performance, Multiprocessors and Multicomputer.

Machine Instructions and Programs: Numbers, Arithmetic Operations and Programs, Instructions and Instruction Sequencing, Addressing Modes, Basic Input/output Operations

INTEL-8086: CPU architecture, Addressing modes - generation of physical address- code segment registers, Zero, one, two, and three address instructions. INTEL 8086 ASSEMBLY LANGUAGE INSTRUCTIONS-Data transfer instructions, input- output instructions, arithmetic, logical, shift, and rotate instructions, Conditional and unconditional transfer.

UNIT-II

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating- Point Numbers and Operations.

Basic Processing Unit: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, Multi-programmed Control.

UNIT-III

The Memory System: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

UNIT-IV

Input/output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

UNIT-V

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets.

Large Computer Systems: Forms of Parallel Processing, Array Processors, The Structure of General- Purpose multiprocessors, Interconnection Networks.

TEXT BOOKS

1. *Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, McGraw Hill Education, 5th Edition, 2013.*
2. *Microprocessors and Interfacing, Douglas Hall, Tata McGraw-Hill.*

REFERENCES

1. *Fundamentals of Computer Organization and Design, Sivarama, P.Dandamudi, Springer Int. Edition.*
2. *Digital Logic & Computer Design, M. Morris Mano, PHI/Pearson Education*
3. *Computer Organization and Architecture, William Stallings, 7th Edition, Pearson/PHI, 2007.*

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
4	-	-	4

(25MC9103) SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *To learn the basic concepts of software engineering and life cycle models*
2. *To explore the issues in software requirements specification and enable to write SRS documents for software development problems*
3. *To elucidate the basic concepts of software design and enable to carry out procedural and object oriented design of software development problems*
4. *To understand the basic concepts of black box and white box software testing and enable to design test cases for unit, integration, and system testing*
5. *To reveal the basic concepts in software project management*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand concepts of software engineering, SDLC models for effective project planning and management using models like COCOMO and risk management techniques.*
2. *Analyze software requirements by applying techniques such as requirement gathering, SRS based on IEEE standards, and using decision tables and formal system development methods.*
3. *Design software systems using principles of cohesion, coupling, control hierarchies, and distinguishing object-oriented and function-oriented approaches.*
4. *Apply coding standards, OO Approaches, conduct thorough code reviews, and perform testing to ensure software quality.*
5. *Evaluate standards (like ISO 9000, CMM), maintenance models, reverse engineering, and estimation for sustainable software development and maintenance.*
6. *Utilize software reuse principles at both the program and organizational levels to improve software development efficiency and maintainability.*

UNIT - I

Basic concepts: abstraction versus decomposition, evolution of software engineering techniques.

Software development life cycle (SDLC) models: Iterative waterfall model, Prototype model, Evolutionary model, Spiral model, RAD model, Agile models,

Software Project Management: project planning, project estimation, COCOMO, Halstead's Software Science, project scheduling, staffing, Organization and team structure, risk management, configuration management.

UNIT - II

The nature of software, The Unique nature of Webapps, Software Myths, Requirements gathering and analysis, software requirements specification, Traceability, Characteristics of a Good SRS Document, IEEE 830 guidelines, representing complex requirements using decision tables and decision trees, overview of formal system development techniques. Axiomatic specification, algebraic specification.

UNIT - III

Good Software Design, Cohesion and coupling, Control Hierarchy: Layering, Control Abstraction, Depth and width, Fan-out, Fan-in, Software design approaches, object oriented vs. function oriented design. Overview of SA/SD methodology, structured analysis, Data flow diagram, Extending DFD technique to real life systems, Basic Object oriented concepts, UML Diagrams, Structured design, Detailed design, Design review, Characteristics of a good user interface, User Guidance and Online Help, Mode-based Vs. Mode-less Interface, Types of user interfaces, Component-based GUI development

User interface design methodology: GUI design methodology.

UNIT - IV

Coding standards and guidelines, code review, software documentation, Testing, Black Box Testing, White Box Testing, debugging, integration testing, Program Analysis Tools, system testing, performance testing, regression testing, Testing Object Oriented Programs.

UNIT - V

Software reliability, Statistical testing, Software quality and management, ISO 9000, SEI capability maturity model (CMM), Personal software process (PSP), Six sigma, Software quality metrics, CASE and its scope, CASE environment, CASE support in software life cycle, Characteristics of software maintenance, Software reverse engineering, Software maintenance processes model, Estimation maintenance cost. Basic issues in any reuse program, Reuse approach, Reuse at organization level.

TEXT BOOKS

1. RajibMall, “Fundamentals of Software Engineering”, 5th Edition, PHI, 2018.
2. Pressman R, “Software Engineering- Practioner Approach”, McGraw Hill.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
4	-	-	4

(25MC9104) DATA STRUCTURES USING C

COURSE OBJECTIVES

1. *To illustrate the basic concepts of C programming language.*
2. *To discuss the concepts of Functions, Arrays, Pointers and Structures.*
3. *To familiarize with Stack, Queue and Linked lists data structures.*
4. *To explain the concepts of non-linear data structures like graphs and trees.*
5. *To learn the different types of searching and sorting techniques.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Demonstrate foundational programming skills in C language including data types, control structures, functions, arrays, pointers, structures, unions, and string handling.*
2. *Understand and implement basic data structures such as stacks and queues, including their operations, applications in expression evaluation, and related algorithms.*
3. *Design and manipulate linked list data structures including singly, doubly, and circular linked lists for dynamic storage management.*
4. *Analyze tree and graph data structures, to perform operations like traversals, and apply graph algorithms such as BFS and DFS to solve computational problems.*
5. *Implement various searching and sorting algorithms for efficient data manipulation.*
6. *Develop proficiency in selecting and applying appropriate data structures and algorithms to real-world problems to optimize performance and resource utilization.*

UNIT - I

Introduction to C Language - C Language Elements, Variable Declarations and Data Types, Operators and Expressions, Decision Statements - If and Switch Statements, Loop Control Statements -while, for, do-while Statements.

Introduction to Functions, Storage classes, Arrays, Structures, Unions, Pointers, Strings and Command line arguments.

UNIT - II

Data Structures, Stacks and Queues: Overview of Data Structure, Representation of a Stack, Stack Related Terms, Operations on a Stack, Implementation of a Stack, Evaluation of Arithmetic Expressions, Infix, Prefix, and Postfix Notations, Evaluation of Postfix Expression, Conversion of Expression from Infix to Postfix, Recursion, Queues - Various Positions of Queue, Representation of Queue, Insertion, Deletion, Searching Operations.

UNIT - III

Linked Lists: Pointers, Singly Linked List, Dynamically Linked Stacks and Queues, Polynomials Using Singly Linked Lists, Using Circularly Linked Lists, Insertion, Deletion

and Searching Operations, Doubly linked lists and its operations, Circular linked lists and its operations.

UNIT - IV

Trees: Tree terminology, representation, Binary tress, representation, Binary tree traversals. Binary Tree Operations, Graphs- Graph terminology, Graph representation, Elementary Graph Operations, Breadth first search (BFS) and Depth first search (DFS), Connected Components, Spanning Trees.

UNIT - V

Searching and Sorting: Sequential, Binary, Exchange (Bubble) Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort. Searching- Linear and Binary Search Methods.

TEXT BOOKS

1. The C Programming Language, Brian W Kernighan and Dennis M Ritchie, Second Edition, Prentice Hall Publication.
2. Fundamentals of Data Structures in C, Ellis Horowitz, SartajSahni, Susan Anderson-Freed, Computer Science Press.
3. Programming in C and Data Structures, J.R.Hanly, Ashok N. Kamthane and A. AnandaRao, Pearson Education.
4. B.A.Forouzon and R.F. Gilberg, "COMPUTER SCIENCE: A Structured Programming Approach Using C", Third edition, CENGAGE Learning, 2016.
5. Richard F. Gilberg&Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach with C", Second Edition, CENGAGE Learning, 2011.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
4	-	-	4

(25MC9105) DATABASE MANAGEMENT SYSTEMS

COURSE OBJECTIVES

1. *Train in the fundamental concepts of database management systems, database modeling and design, SQL, PL/SQL and system implementation techniques.*
2. *Enable students to model ER diagram for any customized application*
3. *Inducting appropriate strategies for optimization of queries.*
4. *Provide knowledge on concurrency techniques*
5. *Demonstrate the organization of Databases*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the concepts of database systems, architecture, data models, database users, and the relational model with schema and keys.*
2. *Write and execute SQL queries involving data definition, manipulation, nested queries, joins, views, transactions, triggers, and advanced SQL features.*
3. *Design database schemas, ER models and convert ER into relational schemas; apply normalization and functional dependencies for relational database.*
4. *Analyze query processing techniques, operation evaluation, query optimization strategies, and assess costs involved in query execution.*
5. *Understand transaction management concepts, atomicity, consistency, isolation, durability, concurrency control mechanisms, and recovery methods for reliability.*
6. *Apply knowledge of locking protocols, deadlock handling, concurrency, and database recovery techniques to ensure robust and efficient database systems.*

UNIT - I

Introduction: Database systems applications, Purpose of Database Systems, view of Data, Database Languages, Database Design, Database Engine, Database and Application Architecture, Database Users and Administrators.

Introduction to Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Algebra

UNIT - II

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub-queries, Modification of the Database. Intermediate SQL: Joint Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Index Definition in SQL, Authorization.

Advanced SQL: Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, Advanced Aggregation Features.

UNIT - III

Database Design and the E-R Model: Overview of the Design Process, The Entity-Relationship Model, Complex Attributes, Mapping Cardinalities, Primary Key, Removing Redundant Attributes in Entity Sets, Reducing E-R Diagrams to Relational Schemas, Extended E-R Features, Entity- Relationship Design Issues, Alternative Notations for Modelling Data, Other Aspects of Database Design.

Relational Database Design: Features of Good Relational Designs, Decomposition Using Functional Dependencies, Normal Forms, Functional-Dependency Theory, Algorithms for Decomposition using Functional Dependencies, Decomposition Using Multivalued Dependencies, More Normal Forms, Atomic Domains and First Normal Form, Database–Design Process, Modelling Temporal Data, Indexing.

UNIT - IV

Query Processing: Overview, Measures of Query cost, Selection Operation, Sorting, Join Operation, Other Operations, Evaluation of Expressions, Query Processing in Memory.

Query optimization: Overview, Transformation of Relational Expressions, Estimating Statistics of Expression Results, Choice of Evaluation Plans, Materialized views, Advanced Topics in Query Optimization.

UNIT - V

Transactions: Transaction Concept, A Simple Transactional Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels, Transactions as SQL Statements.

Concurrency Control: Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Insert Operations. Delete Operations and Predicate Reads, Timestamp-Based Protocols, Validation- Based Protocols, Multiversion Schemes, Snapshot Isolation, Weak Levels of Consistency in Practice, Advanced Topics in Concurrency.

Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Non-Volatile Storage, High Availability Using Remote Backup Systems, Early Lock Release and Logical Undo Operations, ARIES, Recovery in Main-Memory Databases.

TEXT BOOKS

1. A.Silberschatz, H.F.Korth, S.Sudarshan, “Database System Concepts”, 7/e, TMH 2020

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
-	1	2	2

(25MC9106) SOFTWARE ENGINEERING LABORATORY

COURSE OBJECTIVES

1. *To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Develop the ability to analyze a problem statement and prepare comprehensive Software Requirement Specification (SRS) and design documents accurately.*
2. *Create and manage Software Configuration Management and Risk Management documents to support the software development lifecycle.*
3. *Gain proficiency in using design phase CASE tools for system modeling and software design.*
4. *Perform software design tasks effectively through practical application of CASE tools, facilitating accurate representation and documentation.*
5. *Design and develop test cases for unit testing and integration testing to ensure component and system correctness.*
6. *Apply various white-box and black-box testing techniques by designing relevant test cases to verify software functionality and reliability.*

List of Experiments:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
-	1	2	2

(25MC9107) DATA STRUCTURES USING C LABORATORY

COURSE OBJECTIVES

1. *To get familiar with the basic concepts of C programming.*
2. *To design programs using arrays, strings, pointers and structures.*
3. *To illustrate the use of Stacks and Queues*
4. *To apply different operations on linked lists.*
5. *To demonstrate the Binary tree traversal techniques.*
6. *To design searching and sorting techniques*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Develop C programs using recursive and non-recursive functions to solve mathematical and algorithmic problems such as factorial, GCD, and Towers of Hanoi.*
2. *Implement basic operations on arrays, matrices, strings, and pointers to manipulate data effectively and understand parameter passing mechanisms in C.*
3. *Design and implement fundamental data structures such as stacks and queues using arrays and pointers along with expression evaluation and conversion algorithms.*
4. *Create, insert, delete, and traverse various linked list structures (singly, doubly, circular) and understand their dynamic memory management.*
5. *Construct binary trees and perform different tree traversals (preorder, inorder, postorder) to understand hierarchical data representation.*
6. *Implement and analyze searching and sorting algorithms to understand algorithmic efficiency.*

List of Experiments:

1) Write C programs that use both recursive and non-recursive functions
   i) To find the factorial of a given integer.
   ii) To find the GCD (greatest common divisor) of two given integers.
   iii) To solve Towers of Hanoi problem.
2) Write a C program to find both the largest and smallest number in a list of integers.
3) Write a C program that uses functions to perform the following:
   i) Addition of Two Matrices ii) Multiplication of Two Matrices
4) Write a C program that uses functions to perform the following operations:
   i) To insert a sub-string in to a given main string from a given position.
   ii) To delete n Characters from a given position in a given string.
5) Write a C program that displays the position or index in the string S where the string T begins, or – 1 if S doesn‘t contain T.
6) Write a C program to count the lines, words and characters in a given text.
7) Write a C Program to perform various arithmetic operations on pointer variables.
8) Write a C Program to demonstrate the following parameter passing mechanisms:

i) call-by-value ii) call-by-reference.

9) Write a C program that uses functions to perform the following operations:
   i) Reading a complex number ii) Writing a complex number
   iii) Addition of two complex numbers iv) Multiplication of two complex numbers (Note: represent complex number using a structure.)
10) Write C programs that implement stack (its operations) using
   i) Arrays ii) Pointers
11) Write C programs that implement Queue (its operations) using
   i) Arrays ii) Pointers
12) Write a C program that uses Stack operations to perform the following:
   i) Converting infix expression into postfix expression
   ii) Evaluating the postfix expression
13) Write a C program that uses functions to perform the following operations on singly linked list.
   i) Creation ii) Insertion iii) Deletion iv) Traversal
14) Write a C program that uses functions to perform the following operations on Doubly linkedlist.
   i) Creation ii) Insertion iii) Deletion iv) Traversal
15) Write a C program that uses functions to perform the following operations on Circular linkedlist.
   i) Creation ii) Insertion iii) Deletion iv) Traversal
16) Write a C program that uses functions to perform the following:
   i) Creating a Binary Tree of integers
   ii) Traversing the above binary tree in preorder, inorder and postorder.
17) Write C programs that use both recursive and non-recursive functions to perform the following searching operations for a Key value in a given list of integers:
   i) Linear search ii) Binary search
18) Write a C program that implements the following sorting methods to sort a given list of integers in ascending order
   i) Bubble sort ii) Selection sort
19) Write a C program that implements the following sorting methods to sort a given list of integers in ascending order
   i) Insertion sort ii) Merge sort iii) Quick sort

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
-	1	2	2

(25MC9108) DATABASE MANAGEMENT SYSTEMS LABORATORY

COURSE OBJECTIVES

1. *To implement the basic knowledge of SQL queries and relational algebra.*
2. *To construct database models for different database applications.*
3. *To apply normalization techniques for refining of databases.*
4. *To practice various triggers, procedures, and cursors using PL/SQL.*
5. *To design and implementation of a database for an organization*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Design and create database schemas using SQL with appropriate constraints for given applications.*
2. *Use the basics of SQL and construct queries using SQL in database creation and interaction.*
3. *Develop and test stored procedures, functions, triggers, and cursors to enforce business rules.*
4. *Apply normalization techniques to eliminate redundancy and improve database efficiency.*
5. *Demonstrate proficiency in handling complex queries involving joins, subqueries, views, and indexing.*
6. *Use Oracle for PL/SQL Programming, normalizations, and design ER Diagrams for real world applications through case studies.*

List of Experiments:

1. Create a table called Employee with the following structure.

Name	Type
Empno	Number
Ename	Varchar2(20)
Job	Varchar2(20)
Mgr	Number
Sal	Number

   a. Add a column commission with domain to the Employee table.
   b. Insert any five records into the table.
   c. Update the column details of job
   d. Rename the column of Employ table using alter command.
   e. Delete the employee whose empno is19.

2. Create department table with the following structure.

Name	Type
Deptno	Number
Deptname	Varchar2(20)
Location	Varchar2(20)

   a. Add column designation to the department table.
   b. Insert values into the table.
   c. List the records of emp table grouped by dept no.
   d. Update the record where dept no is9.

e. Delete any column data from the table

QUERIES USING DDL AND DML

3. a. Create a user and grant all permissions to the user.
   b. Insert the any three records in the employee table and use rollback. Check the result.
   c. Add primary key constraint and not null constraint to the employee table.
   d. Insert null values to the employee table and verify the result.
4. a. Create a user and grant all permissions to the user.
   b. Insert values in the department table and use commit.
   c. Add constraints like unique and not null to the department table.
   d. Insert repeated values and null values into the table.
5. a. Create a user and grant all permissions to the user.
   b. Insert values into the table and use commit.
   c. Delete any three records in the department table and use rollback.
   d. Add constraint primary key and foreign key to the table.
6. a. Create a user and grant all permissions to the user.
   b. Insert records in the sailor table and use commit.
   c. Add save point after insertion of records and verify save point.
   d. Add constraints not null and primary key to the sailor table.
7. a. Create a user and grant all permissions to the user.
   b. Use revoke command to remove user permissions.
   c. Change password of the user created.
   d. Add constraint foreign key and notnull.
8. a. Create a user and grant all permissions to the user.
   b. Update the table reserves and use save point and rollback.
   c. Add constraint primary key , foreign key and not null to the reserves table
   d. Delete constraint not null to the table column.

QUERIES USING AGGREGATE FUNCTIONS

9. a. By using the group by clause, display the names who belongs to dept no 10 along with average salary.
   b. Display lowest paid employee details under each department.
   c. Display number of employees working in each department and their department number.
   d. Using built in functions, display number of employees working in each department and their department name from dept table. Insert dept name to dept table and insert dept name for each row, do the required thing specified above.
   e. List all employees which start with either B or C.
   f. Display only these ename of employees where the maximum salary is greater than or equal to 5000.
10. a. Calculate the average salary for each different job.
    b. Show the average salary of each job excluding manager.
    c. Show the average salary for all departments employing more than three people.
    d. Display employees who earn more than the lowest salary in department 30
    e. Show that value returned by sign (n)function.
    f. How many days between day of birth to current date
11. a. Show that two substring as single string.
    b. List all employee names, salary and 15% rise in salary.
    c. Display lowest paid emp details under each manager
    d. Display the average monthly salary bill for each deptno.
    e. Show the average salary for all departments employing more than two people.
    f. By using the group by clause, display the eid who belongs to dept no 05 along with average salary.
12. a. Count the number of employees in department20

b. Find the minimum salary earned by clerk.
c. Find minimum, maximum, average salary of all employees.
d. List the minimum and maximum salaries for each job type.
e. List the employee names in descending order.
f. List the employee id, names in ascending order by empid.

13. a. Find the sids ,names of sailors who have reserved all boats called "INTERLAKE Find the age of youngest sailor who is eligible to vote for each rating level with at least two such sailors.
b. Find the sname , bid and reservation date for each reservation.
c. Find the ages of sailors whose name begin and end with B and has at least 3characters.
d. List in alphabetic order all sailors who have reserved red boat.
e. Find the age of youngest sailor for each rating level.
14. a. List the Vendors who have delivered products within 6 months.
b. Display the Vendor details who have supplied both Assembled and Subparts.
c. Display the Sub parts by grouping the Vendor type (Local or Non Local).

PROGRAMS ON PL/SQL

15. a. Write a PL/SQL program to swap two numbers.
b. Write a PL/SQL program to find the largest of three numbers.
16. a. Write a PL/SQL program to find the total & average of 6 subjects and display the grade.
b. Write a PL/SQL program to find the sum of digits in a given umber.
17. a. Write a PL/SQL program to display the number in reverse order.
b. Write a PL/SQL program to check whether the given number is prime or not.
18. a. Write a PL/SQL program to find the factorial of a given number.
b. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius and area.
19. a. Write a PL/SQL program to accept a string and remove the vowels from the string. (When 'hello' passed to the program it should display 'Hll' removing e and o from the worldHello).
b. Write a PL/SQL program to accept a number and a divisor. Make sure the divisor is less than or equal to 10. Else display an error message. Otherwise Display the remainder in words.

PROCEDURES AND FUNCTIONS

20. Write a function to accept employee number as parameter and return Basic +HRA together as single column.
21. Accept year as parameter and write a Function to return the total net salary spent for a given year.
22. Create a function to find the factorial of a given number and hence find NCR.
23. Write a PL/SQL block to pint prime Fibonacci series using local functions.
24. Create a procedure to find the lucky number of a given birth date.
25. Create function to the reverse of given number
26. Create the procedure for palindrome of given number.
27. Create the procedure for GCD: Program should load two registers with two Numbers and then apply the logic for GCD of two numbers. GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisors of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD is found.
28. Write the PL/SQL programs to create the procedure for factorial of given number.
29. Write the PL/SQL programs to create the procedure to find sum of N natural number.
30. Write the PL/SQL programs to create the procedure to find Fibonacci series.
31. Write the PL/SQL programs to create the procedure to check the given number is perfect or not.

CASE STUDY: BOOK PUBLISHING COMPANY

A publishing company produces scientific books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more publications.

A publication covers essentially one of the specialist subjects and is normally written by a single author. When writing a particular book, each author works with on editor, but may submit another work for publication to be supervised by other editors. To improve their competitiveness, the company tries to employ a variety of authors, more than one author being a specialist in a particular subject for the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

CASE STUDY: STUDENT PROGRESS MONITORING SYSTEM

A database is to be designed for a college to monitor students' progress throughout their course of study. The students are reading for a degree (such as BA, BA (Hons) M.Sc., etc) within the framework of the modular system. The college provides a number of modules, each being characterized by its code, title, credit value, module leader, teaching staff and the department they come from. A module is coordinated by a module leader who shares teaching duties with one or more lecturers. A lecturer may teach (and be a module leader for) more than one module. Students are free to choose any module they wish but the following rules must be observed: Some modules require pre-requisites modules and some degree programmes have compulsory modules. The database is also to contain some information about students including their numbers, names, addresses, degrees they read for, and their past performance i.e. modules taken and examination results. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.
3. Create the logical data model i.e., ER diagrams.
4. Comprehend the data given in the case study by creating respective tables with primary keys and foreign keys wherever required.
5. Insert values into the tables created (Be vigilant about Master- Slave tables).
6. Display the Students who have taken M.Sc course
7. Display the Module code and Number of Modules taught by each Lecturer.
8. Retrieve the Lecturer names who are not Module Leaders.
9. Display the Department name which offers ‘English’ module.
10. Retrieve the Prerequisite Courses offered by every Department (with Department names). Present the Lecturer ID and Name who teaches ‘Mathematics’.
11. Discover the number of years a Module is taught.
12. List out all the Faculties who work for ‘Statistics’ Department. List out the number of Modules taught by each Module Leader. List out the number of Modules taught by a particular Lecturer.
13. Create a view which contains the fields of both Department and Module tables. (Hint- The fields like Module code, title, credit, Department code and its name).
14. Update the credits of all the prerequisite courses to 5. Delete the Module ‘History’ from the Module table.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
2	-	-	2

(25MC9109) RESEARCH METHODOLOGY AND IPR

COURSE OBJECTIVES

1. *Identify an appropriate research problem in their interesting domain.*
2. *Understand ethical issues understand the Preparation of a research project thesis report.*
3. *Understand the Preparation of a research project thesis report*
4. *Understand the law of patent and copyrights.*
5. *Understand the Adequate knowledge on IPR*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the formulation of research problems, sources, characteristics of a good research problem, and approaches to investigation including data collection and analysis.*
2. *Develop skills in conducting effective literature reviews, avoid plagiarism, follow research ethics, and prepare technical reports and research proposals according to standardized formats.*
3. *Comprehend the nature of Intellectual Property including patents, designs, trademarks, copyrights, and understand the international framework of IP protection.*
4. *Learn the processes involved in patenting and development, including patent grant procedures, patent cooperation treaty, and technological innovations protection.*
5. *Analyze patent rights, licensing, technology transfer, patent databases, and geographical indications to understand IP commercialization and management.*
6. *Explore recent developments in Intellectual Property Rights (IPR) with focus on emerging areas like biological systems, software, traditional knowledge, and institutional IP policies.*

UNIT - I

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, scope, and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations

UNIT - II

Effective literature studies approaches, analysis Plagiarism, Research ethics, Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee.

UNIT - III

Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

UNIT - IV

Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications.

UNIT - V

New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.

TEXT BOOKS

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students'"
2. Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction"

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
4	-	-	4

(25MC9110) OPERATING SYSTEMS

COURSE OBJECTIVES

1. *Understand basic concepts and functions of operating systems*
2. *Understand the processes, threads and scheduling algorithms.*
3. *Provide good insight on various memory management techniques*
4. *Expose the students with different techniques of handling deadlocks*
5. *Explore the concept of file-system and its implementation issues*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the functions, operations, including computing environments, system structures, disk scheduling and system boot processes in OS.*
2. *Analyze the concepts of processes and threads, apply process scheduling algorithms, and implement inter-process communication.*
3. *Illustrate different memory management schemes, including paging, segmentation, and virtual memory mechanisms.*
4. *Apply appropriate Deadlock detection and prevention techniques, file system organization, RAID and stable storage implementation.*
5. *Examine protection goals, access control models, and system security threats; apply cryptographic methods, user authentication procedures etc.*
6. *Critically evaluate architectural and functional design choices in contemporary operating systems such as Linux and Microsoft Windows.*

UNIT - I

Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Open-Source Operating Systems

System Structures: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Operating system debugging, System Boot.

UNIT - II

Process Concept: Process scheduling, Operations on processes, Inter-process communication, Communication in client server systems.

Multithreaded Programming: Multithreading models, Thread libraries, Threading issues, Examples. Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling, Examples.

Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems - Dining philosophers problem, Readers and writers problem.

UNIT - III

Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation, Examples.

Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation, Examples.

UNIT - IV

Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection And recovery, Deadlock avoidance, Deadlock prevention.

File Systems: Files, Directories, File system implementation, management and optimization. Secondary- Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.

UNIT - V

System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights.

System Security: Introduction, Program threats, System and network threats, Cryptography as a security, User authentication, implementing security defenses, firewalling to protect systems and networks, Computer security classification.

Case Studies: Linux, Microsoft Windows.

TEXT BOOKS

1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2016.
2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008. (Topics: Inter-process Communication and File systems.)

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
4	-	-	4

(25MC9111) DATA MINING AND BUSINESS INTELLIGENCE

COURSE OBJECTIVES

1. *Describing key business intelligence terms.*
2. *Determining the relevance of data to business*
3. *Aligning business intelligence to organizational strategy.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the concepts and lifecycle of data warehousing and business intelligence, articulate their role in transforming raw data into valuable information for decision-making.*
2. *Analyze various architectures of Business Intelligence, Data Warehousing, OLAP operations, and the use of schemas in dimensional data modeling*
3. *Identify the scope and functionalities of Data Mining, categorization and integration and integration of data minings with databases and data warehouses.*
4. *Apply pre-processing techniques to prepare high-quality data for mining ies, and utilize relevant languages and system architectures for data mining tasks*
5. *Demonstrate the process of concept description and perform association rule mining using.*
6. *Evaluate current trends, and develop the ability to assess real-world scenarios requiring BI, DW, and data mining solutions for organizational benefit*

UNIT - I

Overview andconcepts DataWarehousing and Business Intelligence: Why reporting and Analysing data, Raw data to valuable information-Lifecycle of Data - What is Business Intelligence - BI and DW in today's perspective - What is data warehousing - The building Blocks: Defining Features - Data warehouses and data 1marts - Overview of the components - Metadata in the data warehouse - Need for data warehousing - Basic elements of data warehousing - trends in data warehousing.

UNIT - II

The Architecture of BI and DW: BI and DW architectures and its types - Relation between BI and DW - OLAP (Online analytical processing) definitions - Difference between OLAP and OLTP - Dimensional analysis - What are cubes? Drill-down and roll-up - slice and dice or rotation - OLAP models - ROLAP versus MOLAP - defining schemas: Stars, snowflakes and fact constellations

UNIT - III

Introduction to data mining (DM): Motivation for Data Mining - Data Mining-Definition and Functionalities – Classification of DM Systems - DM task primitives - Integration of a Data Mining system with a Database or a Data Warehouse - Issues in DM – KDD Process

UNIT - IV

Data Pre-processing: Why to pre-process data? - Data cleaning: Missing Values, Noisy Data - Data Integration and transformation - Data Reduction: Data cube aggregation, Dimensionality reduction - Data Compression - Numerosity Reduction - Data Mining Primitives - Languages and System Architectures: Task relevant data - Kind of Knowledge to be mined - Discretization and Concept Hierarchy.

UNIT - V

Concept Description and Association Rule Mining: What is concept description? - Data Generalization and summarization-based characterization - Attribute relevance - class comparisons Association Rule Mining: Market basket analysis - basic concepts – Finding frequent item sets: Apriori algorithm - generating rules – Improved Apriori algorithm – Incremental ARM – Associative Classification – Rule Mining.

TEXT BOOKS

1. J. Han, M. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann
2. M. Kantardzic, "Data mining: Concepts, models, methods and algorithms, John Wiley &Sons Inc.
3. Paulraj Ponnian, "Data Warehousing Fundamentals", John Willey.
4. M. Dunham, "Data Mining: Introductory and Advanced Topics", Pearson Education.
5. G. Shmueli, N.R. Patel, P.C. Bruce, "Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner", Wiley India

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
4	-	-	4

(25MC9112) PYTHON PROGRAMMING

COURSE OBJECTIVES

1. *Develop a basic understanding about the Python Concept.*
2. *Apply control structures such as loops and conditional statements.*
3. *Work with data types, functions, modules, and file handling.*
4. *Implement object-oriented programming (OOP) principles in Python*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand Python programming fundamentals, history, features, data types, constants, variables, identifiers, operators, expressions, and syntax.*
2. *Apply decision control statements and iterative constructs to develop logical Python programs and solve problems.*
3. *Develop Python functions using various argument types, lambda functions, recursion, and modular programming concepts.*
4. *Utilize Python's built-in data structures such as lists, tuples, sets, and dictionaries to store, access, manipulate, and manage data efficiently.*
5. *Implement object-oriented programming concepts, using constructors (__init__), and apply inheritance, polymorphism, and method overriding.*
6. *Handle errors and exceptions in Python programs effectively, including raising exceptions and creating user-defined exceptions to ensure robust program execution.*

UNIT - I

Basics of Python Programming: Introduction, History of Python, Features of Python, Writing and Executing Python Program, Constants, Variables and Identifiers – Data Types – Input Operations – Comments – Reserved Words – Indentation.

Operators and Expressions: Arithmetic, Comparison, Assignment, Unary, Bitwise, Shift, Logical, Membership and Identity Operators – Expressions in Python – Operations on Strings

UNIT - II

Decision Control Statements: Introduction – selection or conditional branching if, if-else, if-elif-else, nested if.

Basic Loop Structures or Iterative Statements: for loop, while loop, nested loops, break, continue, pass. Example programs on control flow.
Case Study 1 – Simple Calculator
Case Study 2 – Generating a Calendar

UNIT - III

Function: Definition – Calling - Scope of the Variables in a Function – return statement – Passing Arguments like Required, Keyword, Default and Variable-length Arguments – Lambda Functions – Recursive Functions.

Modules: from import statement – Making your own Modules – the dir() function – Standard Library modules – Packages in python.

UNIT - IV
Data Structures
Lists: Creation – Accessing values – Manipulating values – List Operations.
Tuple: Creation **–** Accessing values – Manipulating values – Tuple Operations.
Sets: Creation **–** Accessing values – Manipulating values – Set Operations
Dictionaries: Creation – Accessing values – Adding, Modifying and Deleting Items – Built-in dictionary methods.

UNIT - V
Object Oriented Programming: Features of OOP – Merits and Demerits of OOP.
Classes and Objects: Defining Classes – Creating Objects – Class Method and self Argument - The __init__() Method – Public and Private – Inheritance – Polymorphism and Method Overriding.
Error and Exception handling: Introduction – Handling Exception – Raising Exceptions – User Defined Exceptions.

TEXT BOOKS
1. *Python Programming: Using Problem Solving Approach*, Reema Thareja, Oxford University Press 2017.
2. *Python Programming: A Modern Approach*, VamsiKurama, Pearson

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(25MC9113) COMPUTER NETWORKS
(Programme Elective – I)

COURSE OBJECTIVES

1. *Introduce the basic concepts of Computer Networks.*
2. *Introduce the layered approach for design of computer networks*
3. *Expose the network protocols used in Internet environment*
4. *Explain the format of headers of IP, TCP and UDP*
5. *Familiarize with the applications of Internet*
6. *Elucidate the design issues for a computer network*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the concepts of the Internet architecture, including the network edge, network core, and the factors affecting delay, loss, and throughput in packet-switched networks.*
2. *Analyze the protocol layers and their service models, along with the evolution and history of computer networking and Internet technologies.*
3. *Demonstrate knowledge of network application principles and protocols such as HTTP, FTP, SMTP, DNS, and peer-to-peer applications, highlighting their roles in the modern Internet.*
4. *Apply the transport-layer services including multiplexing, demultiplexing, connectionless protocols like UDP, connection-oriented protocols like TCP, reliable data transfer mechanisms, and congestion control techniques.*
5. *Illustrate datagram networks, IP addressing and forwarding, routing algorithms, and various routing strategies including broadcast and multicast routing.*
6. *Implement link-layer functionalities, error detection and correction, multiple access protocols, switched LANs, link virtualization, and data center networking, culminating in the ability to interpret real-world web page request processes.*

UNIT - I

What is the Internet, The Network Edge, The Network Core, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol Layers and their Service Models, Networks under attack, History of Computer Networking and the Internet.

UNIT - II

Principles of Network Applications, The web and HTTP, File transfer: FTP, Electronic mail in the internet, DNS-The Internet's Directory Service, Peer-to-Peer Applications.

UNIT - III

Introduction and Transport-Layer Services, Multiplexing and De-multiplexing, Connectionless Transport: UDP, Principles of Reliable Data transfer, Connection-Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control.

UNIT - IV

Introduction, Virtual Circuit and Datagram Networks, The Internet Protocol(IP): Forwarding and Addressing in the Internet, Routing Algorithms, Routing in the Internet, Broadcast and Multicast Routing.

UNIT - V

Introduction to the Link Layer, Error-Detection and Correction Techniques, Multiple Access Links and Protocols, Switched Local Area Networks, Link Virtualization: A Network as a Link Layer, Data Centre Networking, Retrospective: A Day in the Life of a Web Page Request.

TEXT BOOKS

1. James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach", 6th edition, Pearson, 2019.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(25MC9114) SOFTWARE TESTING METHODOLOGIES
(Programme Elective – I)

COURSE OBJECTIVES

1. *To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.*
2. *To develop skills in software test automation and management using latest tools.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the concepts of software testing, purpose, taxonomy, and consequences of bugs, and differentiate key testing dichotomies and models.*
2. *Analyze flow graphs and path testing techniques, including predicates, path sensitizing, and path instrumentation to identify and test program paths effectively.*
3. *Execute transaction flow and dataflow testing strategies to validate transactions and data usage in software, applying domain and interface testing methods to enhance test coverage.*
4. *Apply path products, regular expressions, and logic-based testing tools such as decision tables and KV charts for specification-based testing and anomaly detection.*
5. *Develop and apply state-based testing techniques including creating and evaluating state graphs and transition testing to improve software testability.*
6. *Implement graph matrices and node reduction algorithms for software testing automation, gain practical exposure to testing tools to build test suites efficiently.*

UNIT - I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs.

Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and Achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II

Transaction Flow Testing: Transaction flows, transaction flow testing techniques.

Dataflow Testing: Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability

UNIT - III

Paths, Path products and Regular expressions: Path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

Logic Based Testing: Overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV

State, State Graphs and Transition testing: State graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V

Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

TEXT BOOKS

1. Software Testing Techniques - BarisBeizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(25MC9115) MANAGERIAL ECONOMICS AND FINANCIAL ACCOUNTANCY
(Programme Elective – I)

COURSE OBJECTIVES

1. *To enable the student to understand and appreciate, with a practical insight, the importance of certain basic issues governing the business operations namely: demand and supply, production function, cost analysis, markets, forms of business organizations, capital budgeting and financial accounting and financial analysis.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the concepts of managerial economics, nature and scope of demand analysis, demand determinants, and apply the law of demand with its exceptions and elasticity concepts for effective business decision-making.*
2. *Analyze production functions including ISO quants, ISO costs, laws of returns, and evaluate cost structures, break-even analysis to determine optimal input combinations and cost management in organizations.*
3. *Evaluate various market structures, perfect competition, monopoly, monopolistic competition, and assess pricing methods and policies to formulate effective pricing decisions under diverse market conditions.*
4. *Compare organizational structures such as sole proprietorship, partnership, joint stock companies, public enterprises, and assess the impact of post-liberalization economic changes on business strategies and operations.*
5. *Demonstrate the ability to record financial transactions using double-entry bookkeeping, prepare journals, ledgers, trial balances, and construct final accounts including trading, profit & loss accounts, and balance sheets with simple adjustments.*
6. *Calculate, analyze, interpret key financial ratios related to liquidity and activity, including current ratio, quick ratio, inventory turnover, and debtor turnover, to evaluate the financial health and operational efficiency of a business.*

UNIT - I

Introduction & Demand Analysis: Definition, Nature and Scope of Managerial Economics. Demand Analysis: Demand Determinants, Law of Demand and its exceptions. Elasticity of Demand: Definition, Types, Measurement and Significance of Elasticity of Demand. Demand Forecasting, Factors governing demand forecasting, methods of demand forecasting.

UNIT - II

Production Function- Isoquants and Isocosts, MRTS, Least Cost Combination of Inputs, Cobb-Douglas Production function, Laws of Returns, Internal and External Economies of Scale. Cost Analysis: Cost concepts. Break-even Analysis (BEA)-Determination of Break-Even Point (simple problems) – Managerial Significance.

UNIT - III

Market structures: Types of competition, Features of Perfect competition, Monopoly and Monopolistic Competition. Price-Output Determination in case of Perfect Competition and Monopoly.

Objectives and Policies of Pricing- Methods of Pricing: Cost Plus Pricing. Marginal Cost Pricing, Sealed Bid Pricing, Going Rate Pricing, Limit Pricing, Market Skimming Pricing, Penetration Pricing. Two-Part Pricing, Block Pricing, Bundling Pricing, Peak Load Pricing, Cross Subsidization.

UNIT - IV

Business & New Economic Environment: Characteristic features of Business, Features and evaluation of Sole Proprietorship, Partnership. Joint Stock Company. Public Enterprises and their types, Changing Business Environment in Post-liberalization scenario.

UNIT - V

Introduction to Financial Accounting: Double-Entry Book Keeping, Journal. Ledger. Trial Balance- Final Accounts (Trading Account. Profit and Loss Account and Balance Sheet with simple adjustments).

Financial Analysis through ratios: Computation, Analysis and Interpretation of Liquidity Ratios (Current Ratio and quick ratio). Activity Ratios (Inventory turnover ratio and Debtor Turnover ratio).

TEXT BOOKS

1. Aryasri: Managerial Economics and Financial Analysis, TMH, 2009.
2. Varshney & Maheswari: Managerial Economics, Sultan Chand, 2009.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(25MC9116) OPERATIONS RESEARCH
(Open Elective – I)

COURSE OBJECTIVES

1. *To impart knowledge in concepts and tools of Operations Research*
2. *To understand mathematical models used in Operations Research*
3. *To apply these techniques constructively to make effective business decisions*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Formulate and solve linear programming problems using graphical and simplex methods to optimize real-world decision-making scenarios.*
2. *Analyze and solve dual problems, apply the dual simplex method, and solve transportation, assignment, and traveling salesman problems for operational efficiency.*
3. *Apply game theory concepts to analyze two-person zero-sum games using maxmin principles, mixed strategies, and dominance properties in decision-making.*
4. *Employ CPM and PERT techniques to schedule projects effectively, calculate critical paths, and perform project crashing analyses for optimized project management.*
5. *Understand queuing theory fundamentals, characterize different queuing systems using Poisson and exponential distributions, and analyze M/M/1 and multi-server queue models.*
6. *Develop simulation models including queuing system simulation with event list, utilize pseudorandom number generation techniques, and apply Monte Carlo simulation methods for complex system analysis.*

UNIT – I

Linear programming problems - Mathematical formulation, graphical method of solution, simplex method

UNIT - II

Duality in linear programming problems, dual simplex method, sensitivity analysis, transportation and assignment problems, Traveling Salesman Problem.

UNIT - III

Game theory Introduction, two-person zero-sum games, some basic terms, the maxmin principle, games without saddle points-Mixed Strategies, graphic solution of 2 * n and m*2 games, dominance property.

CPM & PERT- project scheduling, critical path calculations, Crashing.

UNIT - IV

Queuing Theory: Basic structure of queuing systems, roles of the Poisson and exponential distributions, classification of queues basic results of M/M/1: FIFO systems, Extension to multi-server queues.

UNIT - V

Simulation: simulation concepts, simulation of a queuing system using event list, pseudorandom numbers, multiplication congruential algorithm, inverse transformation method, basic ideas of Monte-Carlo simulation.

TEXT BOOKS

1. Taha.H.A ,operation Research : An Introduction, McMilan publishing Co., 1982. 7th ed.
2. Ravindran A, Philips D.T &Solbery.J.J, Operations Research: Principles and practice, John Wiley & Sons, New York, 1987.

REFERENCES

1. Frank S. Budnick, Dennis Mcleavey and Richard Mojena, Principles of Operations Research for Management. All India Traveler Book seller, Delhi.
2. Gillet.B.E., Introduction to Operations Research - A Computer oriented algorithmic approach, McGraw Hill, 1987.
3. Joseph.G.Ecker& Michael KupperSchimd, Introduction to operations Research, John Wiley & Sons, 1988.
4. Hillier.F.S&Liberman.G.J, operation Research, Second Edition, Holden Day Inc, 1974.
5. KantiSwarup, Gupta.P.K. & Man Mohan, operations Research, S.Chand& Sons.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(25MC9117) DIGITAL MARKETING
(Open Elective – I)

COURSE OBJECTIVES

1. *The primary objective of this module is to examine and explore the role and importance of digital marketing in today's rapidly changing business environment.*
2. *It also focuses on how digital marketing can be utilized by organizations and how its effectiveness can measured.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand digital marketing landscape, online market space, digital marketing strategies, and the components involved in building and planning brand websites with effective content marketing.*
2. *Demonstrate the ability to design and implement search engine optimization (SEO) strategies including keyword research, on-page and off-page optimization techniques.*
3. *Apply various email marketing techniques, email automation, lead generation, integration with social media and mobile platforms to measure and maximize campaign effectiveness.*
4. *Utilize mobile marketing channels effectively, location-based services, contextual targeting, coupons, mobile apps, SMS campaigns, and profiling for targeted marketing.*
5. *Leverage social media marketing tools and channels to build brand conversations, create customer engagement, loyalty, and influencer marketing strategies for successful digital campaigns.*
6. *Analyze digital marketing data using web analytics, AdWords, social media analytics, and adapt strategies based on insights with an awareness of digital transformation trends and channel attribution models.*

UNIT – I

Online Market space- Digital Marketing Strategy- Components -Opportunities for building Brand- Website - Planning and Creation- Content Marketing.

UNIT - II

Search Engine optimisation - Keyword Strategy- SEO Strategy - SEO success factors - On-Page Techniques - Off-Page Techniques. Search Engine Marketing- How Search Engine works- SEM components- PPC advertising -Display Advertisement

UNIT - III

E- Mail Marketing - Types of E- Mail Marketing - Email Automation - Lead Generation - Integrating Email with Social Media and Mobile- Measuring and maximising email campaign effectiveness. Mobile Marketing- Mobile Inventory/channels- Location based; Context based; Coupons and offers, Mobile Apps, Mobile Commerce, SMS Campaigns-Profiling and targeting.

UNIT - IV

Social Media Marketing - Social Media Channels- Leveraging Social media for brand conversations and buzz. Successful /benchmark Social media campaigns. Engagement Marketing-Building Customer relationships - Creating Loyalty drivers - Influencer Marketing.

UNIT - V

Digital Transformation & Channel Attribution- Analytics- Ad-words, Email, Mobile, Social Media, Web Analytics - Changing your strategy based on analysis- Recent trends in Digital marketing.

TEXT BOOKS

1. Fundamentals of Digital Marketing by Puneet Singh Bhatia Publisher: Pearson Education; First edition (July 2017)
2. Digital Marketing by Vandana Ahuja; Publisher: Oxford University Press (April 2015)

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(25MC9118) CLOUD COMPUTING
(Open Elective – I)

COURSE OBJECTIVES

1. *To understand the need of Cloud Computing.*
2. *To develop cloud applications.*
3. *To demonstrate design the architecture for new cloud application.*
4. *To teach how to re-architect the existing application for the cloud.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the fundamental concepts, characteristics, service models, and deployment models of cloud computing including virtualization, scalability, elasticity, and cloud security principles.*
2. *Analyze and apply cloud technologies and service platforms such as compute, storage, database, application, content delivery, analytics, and identity & access management services in designing cloud-based solutions.*
3. *Demonstrate the ability to setup and manage Hadoop clusters, implement MapReduce programming model, and design cloud applications using reference architectures and appropriate data storage approaches.*
4. *Develop cloud applications using Python across popular cloud service providers, employing Python libraries, frameworks, and APIs including RESTful web service design.*
5. *Apply big data analytics techniques like clustering, classification, recommendation systems, and multimedia cloud applications such as live video streaming and video transcoding.*
6. *Evaluate cloud security architectures and mechanisms including authentication, authorization, key management, and auditing.*

UNIT – I

Introduction to cloud computing: Introduction, Characteristics of cloud computing, Cloud Models, Cloud Services Examples, Cloud Based services and applications.

Cloud concepts and Technologies: Virtualization, Load balancing, Scalability and Elasticity, Deployment, Replication, Monitoring, Software defined, Network function virtualization, Map Reduce, Identity and Access Management, services level Agreements, Billing.

Cloud Services and Platforms: Compute Services, Storage Services, Database Services, Application services, Content delivery services, Analytics Services, Deployment and Management Services, Identity & and Access Management services, Open Source Private Cloud software

UNIT - II

Hadoop & MapReduce: Apache Hadoop, Hadoop MapReduce Job Execution, Hadoop Schedulers, Hadoop Cluster setup.

Cloud Application Design: Reference Architecture for Cloud Applications, Cloud Application Design Methodologies, Data Storage Approaches.

Python Basics: Introduction, Installing Python, Python data Types & Data Structures, Control flow, Function, Modules, Packages, File handling, Date/Time Operations, Classes.

UNIT - III

Python for Cloud: Python for Amazon web services, Python for Google Cloud Platform, Python for windows Azure, Python for MapReduce, Python packages of Interest, Python web Application Frame work, Designing a RESTful web API.

Cloud Application Development in Python: Design Approaches, Image Processing APP, Document Storage App, MapReduce App, Social Media Analytics App.

UNIT - IV

Big Data Analytics: Introduction, Clustering Big Data, Classification of Big data, Recommendation of Systems.

Multimedia Cloud: Introduction, Case Study: Live video Streaming App, Streaming Protocols, case Study: Video Transcoding App.

Cloud Application Benchmarking and Tuning: Introduction, Workload Characteristics, Application Performance Metrics, Design Considerations for a Benchmarking Methodology, Benchmarking Tools, Deployment Prototyping, Load Testing & Bottleneck Detection case Study, Hadoop benchmarking case Study.

UNIT - V

Cloud Security: Introduction, CSA Cloud Security Architecture, Authentication, Authorization, Identity & Access Management, Data Security, Key Management, Auditing.

Cloud for Industry, Healthcare & Education: Cloud Computing for Healthcare, Cloud computing for Energy Systems, Cloud Computing for Transportation Systems, Cloud Computing for Manufacturing Industry, Cloud computing for Education.

Migrating into a Cloud: Introduction, Broad Approaches to migrating into the cloud, the seven –step model of migration into a cloud.

Organizational readiness and Change Management in The Cloud Age :Introduction, Basic concepts of Organizational Readiness, Drivers for changes : A frame work to comprehend the competitive environment, common change management models, change management maturity models, Organizational readiness self – assessment.

TEXT BOOKS

1. Cloud computing A hands-on Approach‖ By ArshdeepBahga, Vijay Madisetti, Universities Press, 2016
2. Cloud Computing Principles and Paradigms: By Raj kumarBuyya, James Broberg, AndrzejGoscinski, wiley, 2016

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
-	1	2	2

(25MC9119) OPERATING SYSTEMS LABORATORY

COURSE OBJECTIVES

1. *To understand the functionalities of various layers of OSI model*
2. *To explain the difference between hardware, software; operating systems, programs and files.*
3. *Identify the purpose of different software applications.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Implement and simulate various CPU scheduling algorithms and Priority to analyze their impact on process management and CPU utilization.*
2. *Develop C programs to demonstrate synchronization mechanisms by simulating classical concurrency problems using semaphores.*
3. *Apply memory management concepts by simulating contiguous memory allocation techniques as well as Multiprogramming with Variable and Fixed Partitions.*
4. *Analyze and implement* page replacement algorithms *FIFO, LRU, and Optimal to understand their role in virtual memory management.*
5. *Simulate and evaluate various file organization techniques and file allocation strategies to manage file systems efficiently.*
6. *Implement deadlock handling Algorithm for deadlock avoidance and prevention, as well as disk scheduling algorithms to improve overall system performance.*

List of Experiments:

1) Simulate the following CPU scheduling algorithms.
   a) FCFS b) SJF c) Round Robin d) Priority.
2) Write a C program to simulate producer-consumer problem using Semaphores
3) Write a C program to simulate the concept of Dining-philosophers problem.
4) Simulate MVT and MFT.
5) Write a C program to simulate the following contiguous memory allocation Techniques
   a) Worst fit b) Best fit c) First fit.
6) Simulate all page replacement algorithms
   a)FIFO b) LRU c) OPTIMAL
7) Simulate all File Organization Techniques
   a) Single level directory b) Two level directory
8) Simulate all file allocation strategies
   a) Sequential b) Indexed c) Linked.
9) Simulate Bankers Algorithm for Dead Lock Avoidance.
10) Simulate Bankers Algorithm for Dead Lock Prevention.
11) Write a C program to simulate disk scheduling algorithms.
    a) FCFS b) SCAN c) C-SCAN

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
-	1	2	2

(25MC9120) DATA MINING AND BUSINESS INTELLIGENCE LABORATORY

COURSE OBJECTIVES

1. *Describing key business intelligence terms.*
2. *Determining the relevance of data to business*
3. *Aligning business intelligence to organizational strategy.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand and explain the KDD process and its key steps*
2. *Implement association rule mining algorithms to identify frequent itemsets and generate meaningful association rules from transactional datasets.*
3. *Apply classification algorithms using Weka software to build predictive models, evaluate their accuracy, and interpret classification results.*
4. *Use libraries and tools (Weka) to perform data segmentation, understand cluster formation, and analyze underlying data groupings.*
5. *Implement regression analysis on datasets to model relationships between variables and predict numerical outcomes.*
6. *Demonstrate data cleaning, conduct exploratory data analysis, dimensionality reduction and market basket analysis techniques for practical data mining applications.*

List of Experiments:

1) Write a note on KDD process.
2) Write a program to implement apriori algorithm.
3) Demonstration of Association rule process on dataset supermarket.arff using apriori algorithm
4) Write a program to implement any classifier.
5) Write a program to implement any clustering algorithm.
6) Write a program to implement Linear Regression.
7) Write a python program to read data from a csv file, perform simple data analysis and generate basic insights.(use pandas in python libraries).
8) Write a python program to implement data cleaning.
9) Write a program to implement dimensionality reduction.
10) Write a program to implement market-basket analysis.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
-	1	2	2

(25MC9121) PYTHON PROGRAMMING LABORATORY

COURSE OBJECTIVES

1. *Develop a basic understanding about the Python Concept.*
2. *Apply control structures such as loops and conditional statements.*
3. *Work with data types, functions, modules, and file handling.*
4. *Implement object-oriented programming (OOP) principles in Python*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand and implement basic Python programming concepts to write efficient and effective programs.*
2. *Demonstrate skills to manipulate and process text files, extraction and analysis of data.*
3. *Design functions in Python to perform mathematical computations.*
4. *Analyze Python's built-in modules to implement diverse functionalities and solve real-world problems.*
5. *Acquire practical knowledge of Python data and apply various operations on them effectively.*
6. *Apply object-oriented programming concepts such as inheritance and polymorphism, along with exception handling for robust program design.*

List of Experiments:

1) Write a python program that takes in command line arguments as input and print the number of arguments.
2) Write a python program to find the most frequent words in a text file.
3) Write a python program find the square root of a number using functions.
4) Write a python program to find factorial of a number using functions.
5) Write a python program to demonstrate the modules like math, random and datetime.
6) Write a python program to perform various operations on List Data Structure.
7) Write a python program to perform various operations on Set Data Structure.
8) Write a python program to perform various operations on Tuple Data Structure.
9) Write a python program to perform various operations on Dictionary Data Structure.
10) Write a python program to demonstrate the concept of Inheritance.
11) Write a python program to demonstrate the concept of Polymorphism.
12) Write a python program to demonstrate Exception Handling.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
1	-	2	2

(25MC9122) JAVA PROGRAMMING
(Skill Oriented Course – I)

COURSE OBJECTIVES

1. *Study the syntax, semantics and features of Java Programming Language*
2. *Study the Object-Oriented Programming Concepts of Java Programming language*
3. *Learn the method of creating Multi-threaded programs and handle exceptions*
4. *Learn Java features to create GUI applications & perform event handling*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the fundamentals of Java programming language like keywords, data types, operators, Applets and control structures.*
2. *Analyze core Object-Oriented Programming (OOP) concepts such as data abstraction, encapsulation, inheritance, polymorphism, classes, and objects.*
3. *Implement Java classes, objects, methods, and arrays including multidimensional arrays and strings.*
4. *Apply inheritance concepts and method overriding, including abstract classes in complex problems.*
5. *Use byte and character streams for file I/O operations multithreaded programming concepts and applets.*
6. *Develop GUI applications using Swing components, containers, layout managers, and implement event handling in Java applications.*

UNIT – I

The Java Language, OOPS Concepts:-Data abstraction, encapsulation, inheritance, Benefits of Inheritance, Polymorphism, classes and objects, Procedural and object oriented programming paradigms. JDK, simple program, Java keywords, identifiers in java, introducing data types and operators, program control structures.

UNIT - II

Introducing classes, objects, and methods, Arrays, multidimensional arrays, strings, a closer look at methods and classes. Inheritance: Inheritance hierarchies super and sub classes, Member access rules, super keyword, preventing inheritance: final classes and methods, the Object class and its methods. Polymorphism: Dynamic binding, method overriding, abstract classes and methods.

UNIT - III

Interface fundamentals, creating and implementing an interface, using interface references, implementing multiple interfaces, constants in interfaces, interfaces can be extended, packages, Exception handling.

UNIT - IV

Byte streams and character streams, byte and character stream classes, using byte streams for reading and writing, reading and writing binary data, Multi-threaded programming, Applet basics, a complete applet skeleton, applet initialization and termination, requesting repainting.

UNIT - V

Swings – the origin and design philosophy of swing, components and containers, layout managers, event handling.

TEXT BOOKS

1. "Java Fundamentals A Comprehensive Introduction" Herbert Schildt and Dale Skrien, McGraw Hill.
2. "Java – How to Program", Paul Deitel, Harvey Deitel, PHI

REFERENCES

1. "Programming with Java" T.V.Suresh Kumar, B.Eswara Reddy, P.Raghavan Pearson Edition.
2. "Core Java", Nageswar Rao, Wiley Publishers.
3. "Thinking in Java", Bruce Eckel, Pearson Education.
4. "Programing In java", Malhotra, Oxford University Press

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
4	-	-	4

(25MC9124) WEB TECHNOLOGIES

COURSE OBJECTIVES

1. *To introduce PHP language for server-side scripting*
2. *To introduce XML and processing of XML Data with Java*
3. *To introduce Server-side programming with Java Servlets and JSP*
4. *To introduce Client-side scripting with Javascript and AJAX.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Demonstrate the ability to declare and manipulate variables, data types, arrays, strings, operators, control structures, and functions using PHP for web development.*
2. *Build web forms in PHP, handle file uploads, connect to MySQL databases, execute simple queries, and manage sessions, cookies, and basic file operations.*
3. *Design dynamic web pages using HTML common tags, tables, forms, frames, and style them with Cascading Style Sheets (CSS).*
4. *Develop and validate XML documents with proper tag definitions, attributes, DTDs, XML Schemas, and parse XML data using DOM and SAX parsers.*
5. *Develop Java web applications using Servlets, including handling HTTP requests and responses, session management, initialization parameters, and database connectivity with JDBC.*
6. *Create interactive and dynamic web interfaces using JavaScript, including variable declaration, functions, event handling, DOM manipulation, and form validation.*

UNIT – I

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies

File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

UNIT - II

HTML: Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;

XML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemes, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

UNIT - III

Introduction to Servlets: Common Gateway Interface (CGI), Life cycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

UNIT - IV

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

UNIT - V

Client-side Scripting: Introduction to JavaScript, JavaScript language – declaring variables, scope of variables, functions, event handlers (on click, on submit etc.), Document Object Model, Form validation.

TEXT BOOKS

1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP — Steven Holzner, Tata McGraw-Hil

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
4	-	-	4

(25MC9125) BIG DATA TECHNOLOGIES

COURSE OBJECTIVES

1. *To understand the specialized aspects of big data including big data application, and big data analytics.*
2. *To study different types Case studies on the current research and applications of the Hadoop and big data in industry.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the fundamentals of Big Data, significance, trends, and applications in various domains, with an introduction to Hadoop and Big Data technologies.*
2. *Analyze the concepts of NoSQL, Big Data and aggregate data models, key-value stores, graph databases, replication and sharding techniques, and MapReduce.*
3. *Demonstrate knowledge of Hadoop ecosystem, HDFS architecture, data formats, Hadoop streams and pipes, data integrity, compression, and serialization techniques for effective Big Data processing.*
4. *Develop and execute MapReduce programs, understand MapReduce job workflows, unit testing with MRUnit, job scheduling, task execution, handling failures in MapReduce and YARN, and comprehend different input/output formats.*
5. *Implement Big Data solutions such as HBase and Cassandra, understanding their data models, client APIs, integration with Hadoop, and practical use cases.*
6. *Use Hive for Big Data querying and manipulation through HiveQL, file formats, and query execution to efficiently manage and analyze large-scale datasets.*

UNIT – I

Introduction And Fundamentals of 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 - II

Introduction To Nosql Databases And Big Data Models: NoSQL, aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schemaless databases, materialized views, distribution models, sharding, master-slave replication, peer-peer replication, sharding and replication, consistency, relaxing consistency, version stamps, map-reduce, partitioning and combining, composing MapReduce calculations

UNIT - III

Hadoop Ecosystem And Data Processing: Data format, analysing 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, compression, serialization, Avro, file-based data structures

UNIT - IV

Map Reduce Programming And Workflow Execution: MapReduce workflows, unit tests with MRUnit, test data and local tests, anatomy of MapReduce job run, classic Map-reduce, YARN, failures in classic Map-reduce and YARN, job scheduling, shuffle and sort, task execution, MapReduce types, input formats, output formats.

UNIT - V

Bigdata Storage Solutions: Hbase, data model and implementations, Hbase clients, Hbase examples, praxis. Cassandra, Cassandra data model, Cassandra examples, Cassandra clients, Hadoop integration, Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation, HiveQL queries.

TEXT BOOKS

1. Big Data Analytics, Introduction to Hadoop, Spark, and Machine-Learning, Raj kamal, PreetiSaxena, McGraw Hill, 2018.
2. Big Data, Big Analytics: Emerging Business intelligence and Analytic trends for Today's Business, Michael Minelli, Michelle Chambers, and AmbigaDhiraj, John Wiley & Sons, 2013

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
4	-	-	4

(25MC9126) DEVOPS & AGILE PROGRAMMING

COURSE OBJECTIVES

1. *To give strong knowledge of Agile practices*
2. *To give strong foundation of applications of DevOps*
3. *To give strong foundation of development and its operations*
4. *To give strong foundation of the source code management*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the concepts of DevOps, Agile methodology, practices, Techniques and automation of traditional software development.*
2. *Analyze and implement collaborative Agile practices including pair programming, stand-up meetings, iteration demos, coding standards, and effective communication with customers.*
3. *Demonstrate the ability to conduct root-cause analysis, retrospectives, and performance optimization in Agile projects to enhance process and product quality.*
4. *Apply Agile development techniques such as incremental requirements gathering, test-driven development, customer testing, refactoring, and incremental design to develop high-quality software.*
5. *Conduct root-cause analysis, retrospectives, and continuous process improvement to identify challenges and optimize team performance in Agile projects.*
6. *Implement key DevOps practices, including application deployment, continuous integration, and continuous delivery, to support rapid, reliable, and repeatable software releases.*

UNIT – I

Foundations Of Agile Methodology: Why Agile?, How to be Agile, Understanding XP, Values and Principles, Improve the Process, Eliminate Waste, Deliver Value. Practicing XP-Thinking, Pair Programming, Energized Work, Informative Workspace, Root-Cause Analysis, Retrospectives, Collaborating, Sit Together, Real Customer Involvement, Ubiquitous Language, Stand-Up Meetings, Coding Standards, Iteration Demo, Reporting.

UNIT - II

Agile Practices For Quality And Planning: Releasing-Done Done, No Bugs, Version Control, Ten-Minute Build, Continuous Integration, Collective Code Ownership, Documentation. Planning-Vision, Release Planning, Risk Management, Iteration Planning, Stories, Estimating.

UNIT - III

Agile Development Techniques: Developing-Incremental Requirements, Customer Tests, Test- Driven Development, Refactoring, Incremental Design and Architecture, Spike Solutions, Performance Optimization.

UNIT - IV

Deifinition & Purpose of DevOps: Introduction to DevOps - DevOps and Agile, Minimum Viable Product - Application Deployment - Continuous Integration - Continuous Delivery

UNIT - V

DevOps Culture And Automation (CAMS): CAMS (CULTURE, AUTOMATION, MEASUREMENT AND SHARING): CAMS – Culture - CAMS – Automation - CAMS – Measurement - CAMS – Sharing - Test-Driven Development - Configuration Management - Infrastructure Automation - Root Cause Analysis – Blamelessness - Organizational Learning.

TEXT BOOKS

1. James Shore and Shane Warden, " The Art of Agile Development", O'REILLY, 2007.
2. Robert C. Martin, "Agile Software Development, Principles, Patterns, and Practices", PHI, 2002.
3. The DevOps Handbook - by Gene Kim, Jez Humble, Patrick Debois, and Willis

REFERENCES

1. What is DevOps? - by Mike Loukides
2. The DevOps Handbook - by John Willis, Patrick Debois, Jez Humble, Gene Kim.
3. DevOps: A Software Architect's Perspective - by Len Bass, Ingo Weber, Liming Zhu.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(25MC9127) SOFTWARE ARCHITECTURE & DESIGN PATTERNS
(Program Elective – II)

COURSE OBJECTIVES

1. *Learn How to add functionality to designs while minimizing complexity.*
2. *What code qualities are required to maintain to keep code flexible?*
3. *To understand the common design patterns.*
4. *To explore the appropriate patterns for design problems*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand and explain fundamental software architecture concepts, including the architecture business cycle, architectural styles, patterns, reference models, and quality attributes.*
2. *Analyze and evaluate software architectures using established methods such as ATAM and CBAM, and make informed architectural design decisions.*
3. *Apply structural and creational design patterns such as abstract factory, builder, singleton, adapter, facade, and composite to solve common software design problems effectively.*
4. *Implement behavioral design patterns, including chain of responsibility, command, observer, state, strategy, and template method, to address dynamic aspects of software design.*
5. *Document and communicate software architectures comprehensively using architectural views, design documentation, and reconstruction techniques.*
6. *Apply architectural and design principles through case studies, understanding challenges such as interoperability, high availability, product line development, and user interface design in real-world scenarios.*

UNIT – I

Introduction To Software Architecture And Design Principles: Envisioning Architecture The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views. Creating an Architecture Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT - II

Architecture Evaluation And Decision Making Models: Analyzing Architectures Architecture Evaluation, Architecture design decision making, ATAM, CBAM Moving from One System to Many Software Product Lines, Building systems from off the shelf components, Software architecture in future

UNIT - III

Structural And Creational Design Patterns: Patterns Pattern Description, Organizing catalogs, role in solving design problems, Selection and usage. Creational and Structural

Patterns Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight.

UNIT - IV

Behavioral Design Patterns: Behavioral Patterns Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

UNIT - V

Case Studies In Software Architecture Design: Case Studies A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development. A Case Study (Designing a Document Editor): Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation.

TEXT BOOKS

1. Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick Kazman, Pearson Education, 2003.
2. Design Patterns, Erich Gamma, Pearson Education, 1995.

REFERENCES

1. Beyond Software architecture, Luke Hohmann, Addison wesley, 2003.
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001.
3. Software Design, David Budgen, second edition, Pearson education, 2003
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.
5. Design Patterns in Java, Steven John Metsker& William C. Wake, Pearson education, 2006
6. J2EE Patterns, Deepak Alur, John Crupi& Dan Malks, Pearson education, 2003.
7. Pattern Oriented Software Architecture, F.Buschmann& others, John Wiley & Sons.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(25MC9128) NETWORK SECURITY
(Program Elective – II)

COURSE OBJECTIVES

1. *Network security using various cryptographic algorithms.*
2. *Underlying network security applications. It also focuses on the practical applications that have been implemented and are in use to provide email and web security.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand fundamental security concepts, types of security attacks, services, and mechanisms.*
2. *Analyze the use of symmetric key cryptography principles including block ciphers like DES and IDEA, their operational modes, rounds, and key expansion.*
3. *Apply hash functions, public key cryptography algorithms, their construction, usage, and key processes such as encryption, decryption, and key generation.*
4. *Analyze password-based and cryptographic authentication methods, understanding challenges in distributed systems, password storage, protocols involving key distribution centers (KDCs), and user verification techniques.*
5. *Evaluate security policies, their levels, and user-related security issues, as well as protocol vulnerabilities including reflection attacks, nonce, timestamps, and mutual authentication mechanisms.*
6. *Implement advanced authentication mechanisms involving shared secret and public key protocols, session keys, and one- and two-way public key-based authentication to ensure secure communication.*

UNIT – I

Fundamentals Of Security Mechanisms And Attacks: Attacks, Services and Mechanisms, Security Attacks, Security Services, Integrity check, digital Signature, authentication, has algorithms.

UNIT - II

Symmetric Key Cryptography And Block Ciphers: Block Encryption, DES rounds, S-Boxes IDEA: Overview, comparison with DES, Key expansion, IDEA rounds, Uses of Secret key Cryptography; ECB, CBC, OFB, CFB, Multiple encryptions DES

UNIT - III

Hash Functions And Public Key Cryptography: Length of hash, uses, algorithms (MD2, MD4, MD5, SHS) MD2: Algorithm (Padding, checksum, passes.) MD4 and 5: algorithm (padding, stages, digest computation.) SHS: Overview, padding, stages. Algorithms, examples, Modular arithmetic (addition, multiplication, inverse, and exponentiation) RSA: generating keys, encryption and decryption. Other Algorithms: PKCS, Diffie-Hellman, El-Gamal signatures, DSS, Zero-knowledge signatures.

UNIT - IV

Password Based And Cryptographic Authentication: Password Based, Address Based, Cryptographic Authentication. Passwords in distributed systems, on-line vs offline guessing, storing. Cryptographic Authentication: passwords as keys, protocols, KDC's Certification Revocation, Interdomain, groups, delegation. Authentication of People: Verification techniques, passwords, length of passwords, password distribution, smart cards, biometrics.

UNIT - V

Security Policies And Advanced Authentication Mechanisms: What is security policy, high and low level policy, user issues? Protocol problems, assumptions, Shared secret protocols, public key protocols, mutual authentication, reflection attacks, use of timestamps, nonce and sequence numbers, session keys, one-and two-way public key based authentication.

TEXT BOOKS

1. AtulKahate, Cryptography and Network Security, McGraw Hill.
2. Kaufman, c., Perlman, R., and Speciner, M., Network Security, Private Communication in a public world, 2nd ed., Prentice HallPTR., 2002.
3. Stallings W.Cryptography and Network Security: Principles and Practice, 3rd ed., Prentice Hall PTR.,2003

REFERENCES

1. Stallings, W. Network security Essentials: Applications and standards, Prentice Hall, 2000.
2. Cryptography and Network Security; McGraw Hill; Behrouz A Forouzan.
3. Information Security Intelligence Cryptographic Principles and App. CalabresThomson.
4. Securing A Wireless Network Chris Hurley SPD.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(25MC9129) MACHINE LEARNING
(Program Elective – II)

COURSE OBJECTIVES

1. *This course explains machine learning techniques such as decision tree learning, Bayesian learning etc.*
2. *To understand computational learning theory.*
3. *To study the pattern comparison techniques.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the concepts of machine learning including designing learning systems, concept learning, version spaces, candidate elimination, and decision tree algorithms with their inductive biases.*
2. *Demonstrate artificial neural networks, including perceptrons, multilayer networks, and backpropagation, and apply these to pattern recognition problems.*
3. *Apply Bayesian learning principles, maximum likelihood estimation, Bayesian classifiers, belief networks, and computational learning theory including sample complexity and mistake bounds.*
4. *implement instance-based learning techniques such as k-nearest neighbors, locally weighted regression, and case-based reasoning, distinguishing between lazy and eager learning.*
5. *Utilize evolutionary algorithms including genetic algorithms and genetic programming for hypothesis space search, and implement reinforcement learning methods such as Q-learning and temporal difference learning.*
6. *Incorporate analytical learning methods including explanation-based learning (EBG), use of prior knowledge for search optimization, and hybrid approaches combining inductive and analytical learning.*

UNIT – I

Introduction: Well-posed learning problems, designing a learning system, Perspectives and issues in machine learning Concept learning and the general to specific ordering – introduction, a concept learning task, concept learning as search, find-S: finding a maximally specific hypothesis, version spaces and the candidate elimination algorithm, remarks on version spaces and candidate elimination, inductive bias. Decision Tree Learning – Introduction, decision tree representation, appropriate problems for decision tree learning, the basic decision tree learning algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning

UNIT - II

Artificial Neural Networks-1: Introduction, neural network representation, appropriate problems for neural network learning, perceptions, multilayer networks and the back-propagation algorithm.

Artificial Neural Networks-2: Remarks on the Back-Propagation algorithm, An illustrative example: face recognition, advanced topics in artificial neural networks.

Evaluation Hypotheses: Motivation, estimation hypothesis accuracy, basics of sampling theory, a general approach for deriving confidence intervals, difference in error of two hypotheses, comparing learning algorithms.

UNIT - III

Bayesian learning: Introduction, Bayes theorem, Bayes theorem and concept learning, Maximum Likelihood and least squared error hypotheses, maximum likelihood hypotheses for predicting probabilities, minimum description length principle, Bayes optimal classifier, Gibs algorithm, Naïve Bayes classifier, an example: learning to classify text, Bayesian belief networks, the EM algorithm.

Computational learning theory: Introduction, probably learning an approximately correct hypothesis, sample complexity for finite hypothesis space, sample complexity for infinite hypothesis spaces, the mistake bound model of learning.

Instance-Based Learning: Introduction, k-nearest neighbour algorithm, locally weighted regression, radial basis functions, case-based reasoning, remarks on lazy and eager learning

UNIT - IV

Genetic Algorithms: Motivation, Genetic algorithms, an illustrative example, hypothesis space search, genetic programming, models of evolution and learning, parallelizing genetic algorithms.

Learning Sets of Rules: Introduction, sequential covering algorithms, learning rule sets: summary, learning First-Order rules, learning sets of First-Order rules: FOIL, Induction as inverted deduction, inverting resolution.

Reinforcement Learning: Introduction, the learning task, Q–learning, non-deterministic, rewards and actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.

UNIT - V

Analytical Learning-1: Introduction, learning with perfect domain theories: PROLOG-EBG, remarks on explanation-based learning, explanation-based learning of search control knowledge.

Analytical Learning-2: Using prior knowledge to alter the search objective, using prior knowledge to augment search operators. Combining Inductive and Analytical Learning – Motivation, inductive-analytical approaches to learning, using prior knowledge to initialize the hypothesis.

TEXT BOOKS

1. Machine Learning – Tom M. Mitchell, - MGH

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(25MC9130) MOBILE APPLICATION DEVELOPMENT
(Program Elective – III)

COURSE OBJECTIVES

1. *To demonstrate their understanding of the fundamentals of Android operating systems*
2. *To improves their skills of using Android software development tools*
3. *To demonstrate their ability to develop software with reasonable complexity on mobile platform*
4. *To demonstrate their ability to deploy software to mobile devices*
5. *To demonstrate their ability to debug programs running on mobile devices*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the Android OS architecture, features, development framework, and application lifecycle, including managing activity states and runtime configuration changes.*
2. *Design Android user interfaces using various layouts and UI components, implement event handling, and manage fragments and multi-screen activities effectively.*
3. *Utilize Intents for inter-component communication, manage implicit and explicit intents, and implement Broadcast Receivers and notifications within Android applications.*
4. *Implement persistent storage in Android through file handling using application-specific directories for creating, reading, and managing files.*
5. *Use Shared Preferences to store and retrieve lightweight, key-value pair data efficiently for application settings and user preferences.*
6. *Develop Android applications using SQLite databases, including database creation, table management, data insertion, retrieval, update, and deletion, along with content provider registration and usage.*

UNIT – I

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes. Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes.

UNIT - II

Android User Interface: Measurements – Device and pixel density independent measuring units Layouts – Linear, Relative, Grid and Table Layouts, User Interface (UI) Components – Editable and non-editable Text Views, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers Event Handling – Handling clicks or changes of various UI components Fragments – Creating fragments, Lifecycle of fragments, Fragment states,

Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

UNIT - III

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity Notifications – Creating and Displaying notifications, Displaying Toasts

UNIT - IV

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference

UNIT - V

Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and etindelg data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

TEXT BOOKS

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox)2012
2. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(25MC9131) INTERNET OF THINGS
(Program Elective – III)

COURSE OBJECTIVES

1. *Introduce the fundamental concepts of IoT and physical computing*
2. *Expose the student to a variety of embedded boards and IoT Platforms*
3. *Create a basic understanding of the communication protocols in IoT communications.*
4. *Familiarize the student with application program interfaces for IoT.*
5. *Enable students to create simple IoT applications.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the fundamental concepts of the Internet of Things, including its technology, design principles for connected devices, and the significance of calm and ambient technology in IoT design.*
2. *Describe embedded devices commonly used in IoT such as Arduino, Raspberry Pi, and mobile platforms, and understand always-on IoT concepts with plug computing.*
3. *Demonstrate knowledge of IoT communication protocols including IP addressing, MAC addressing, TCP/UDP ports, and application layer protocols, along with API-based prototyping for real-time IoT interactions.*
4. *Analyze various IoT business models, including the business model canvas, funding strategies, and the lean startup approach for IoT ventures.*
5. *Apply manufacturing process for IoT devices, including designing kits, printed circuit boards, mass production, certification, and scaling software solutions.*
6. *Evaluate ethical considerations in IoT related to privacy, control, environmental impact, and propose solutions to address these challenges.*

UNIT – I

Overview of IoT: The Internet of Things: An Overview, The Flavor of the Internet of Things, The "Internet" of "Things", The Technology of the Internet of Things, Enchanted Objects, Who is Making the Internet of Things? Design Principles for Connected Devices: Calm and Ambient Technology, Privacy, Web Thinking for Connected Devices, Affordances. Prototyping: Sketching, Familiarity, Costs Vs Ease of Prototyping, Prototypes and Production, Open source Vs Close source, Tapping into the community.

UNIT - II

Embedded Devices: Electronics, Embedded Computing Basics, Arduino, Raspberry Pi, Mobile phones and tablets, Plug Computing: Always-on Internet of Things

UNIT - III

Communication in the IoT: Internet Communications: An Overview, IP Addresses, MAC Addresses, TCP and UDP Ports, Application Layer Protocols Prototyping Online Components: Getting Started with an API, Writing a New API, Real-Time Reactions, Other Protocols Protocol

UNIT - IV

Business Models: A short history of business models, The business model canvas, Who is the business model for, Models, Funding an Internet of Things startup, Lean Startups. Manufacturing: What are you producing, Designing kits, Designing printed circuit boards.

UNIT - V

Manufacturing continued: Manufacturing printed circuit boards, Mass-producing the case and other fixtures, Certification, Costs, Scaling up software. Ethics: Characterizing the Internet of Things, Privacy, Control, Environment, Solutions.

TEXT BOOKS

1. Adrian McEwen, Hakim Cassimally - Designing the Internet of Things, Wiley Publications, 2012

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(25MC9132) BLOCKCHAIN TECHNOLOGIES
(Program Elective – III)

COURSE OBJECTIVES

1. *This course is intended to study the basics of Block chain technology. During this course learner will explore various aspects of Block chain technology like application in various domains.*
2. *By implementing learner will have idea about private and public Block chain, and smart contract*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Demonstrate the fundamental concepts of blockchain technology, architecture, transaction and block structure, cryptographic mechanisms such as digital signatures and hashing, and differences between private and public blockchains.*
2. *Analyse the workings of Bitcoin and other cryptocurrencies, including the mining process, wallet usage, decentralization, hard forks, and the Ethereum Virtual Machine with related concepts like Merkle trees and the double-spend problem.*
3. *Understand Ethereum platform fundamentals, including consensus mechanisms, Ethereum accounts, transactions, smart contracts, and development tools such as Metamask for interacting with the blockchain.*
4. *Apply Hyperledger as a blockchain enterprise solution, exploring distributed ledger technology, its challenges, and components.*
5. *Identify and evaluate real-world blockchain applications across domains such as Internet of Things, medical record management, domain name services, and emerging altcoins.*
6. *Use knowledge of blockchain technologies to assess their impact on secure digital transactions and decentralized applications, including development and deployment of smart contracts and understanding blockchain's role in evolving digital economies.*

UNIT – I

Introduction of Cryptography and Block chain: What is Block chain, Block chain Technology Mechanisms &Networks, Block chain Origins, Objective of Block chain, Block chain Challenges, Transactions And Blocks, P2P Systems, Keys As Identity, Digital Signatures, Hashing, and public key cryptosystems, private vs. public Block chain

UNIT - II

Bit Coin and Crypto currency: What is Bitcoin, The Bitcoin Network, The Bitcoin Mining Process, Mining Developments, Bitcoin Wallets, Decentralization and Hard Forks, Ethereum Virtual Machine (EVM), Merkle Tree, Double-Spend Problem, Blockchain And Digital Currency, Transactional Blocks, Impact Of Block chain Technology On Crypto currency.

UNIT - III

Introduction to Ethereum: What is Ethereum, Introduction to Ethereum, Consensus Mechanisms, How Smart Contracts Work, Metamask Setup, Ethereum Accounts, Receiving Ether's What's a Transaction?, Smart Contracts.

UNIT - IV

Introduction to Hyper ledger: What is Hyper ledger? Distributed Ledger Technology & its Challenges, Hyper ledger & Distributed Ledger Technology, Hyper ledger Fabric, Hyper ledger Composer.

UNIT - V

Block chain Applications: Internet of Things, Medical Record Management System, Domain Name Service and Future of Block chain, Alt Coins.

TEXT BOOKS

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven
2. Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive
3. Introduction, Princeton University Press (July 19, 2016).

REFERENCES

1. Antonopoulos, Mastering Bitcoin.
2. Antonopoulos and G. Wood, Mastering Ethereum.
3. D. Drescher, Blockchain Basics. Apress, 2017.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
-	1	2	2

(25MC9133) WEB TECHNOLOGIES LABORATORY

COURSE OBJECTIVES

1. *Understand the web technologies to create adaptive web pages for web application.*
2. *Use CSS to implement a variety of presentation effects to the web application*
3. *Know the concept and implementation of cookies as well as related privacy concerns*
4. *Develop a sophisticated web application that employs the MVC architecture.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Develop and implement PHP scripts that perform fundamental programming tasks including prime number generation, string manipulation, array operations, and file handling.*
2. *Design and create static web pages using HTML to simulate an online bookstore with multiple interconnected pages such as home, registration, user profile, catalog, shopping cart, payment, and order confirmation.*
3. *Apply JavaScript techniques to validate user inputs on various web forms including registration, login, profile management, and payment processing to enhance client-side interactivity and data integrity.*
4. *Create and manage XML documents on the server, including storing user data and implementing query mechanisms to retrieve and display user information dynamically.*
5. *Configure and utilize the Apache Tomcat web server to convert static HTML pages into dynamic web applications using Java Servlets and cookie management for individual user session tracking.*
6. *Develop database-driven dynamic web applications using JSP in adherence to the Model-View-Controller (MVC) architecture, integrating backend user and book catalogs with dynamic content rendering and session management.*

List of Experiments:

1. Write a PHP script to print prime numbers between 1-50.
2. PHP script to
    a. Find the length of a string.
    b. Count no of words in a string.
    c. Reverse a string.
    d. Search for a specific string.
3. Write a PHP script to merge two arrays and sort them as numbers, in descending order.
4. Write a PHP script that reads data from one file and write into another file.
5. Develop static pages (using Only HTML) of an online book store. The pages should resemble: www.amazon.com. The website should consist the following pages.
    a. Home page

b. Registration and user Login
c. User Profile Page
d. Books catalog
e. Shopping Cart
f. Payment By credit card
g. Order Conformation

6. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
7. Create and save an XML document on the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.
8. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
9. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
-	1	2	2

(25MC9134) BIG DATA TECHNOLOGIES LABORATORY

COURSE OBJECTIVES

1. *Apply quantitative modeling and data analysis techniques to the solution of real-world business problems, communicate findings, and effectively present results using data visualization techniques.*
2. *Apply principles of Data Science to the analysis of business problems.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Install and configure Hadoop and Spark environments on single-node clusters and cloud instances (e.g., AWS CentOS 7), enabling hands-on practice with distributed computing platforms.*
2. *Design and implement distributed MapReduce applications to process large datasets such as system logs and weather data, demonstrating skills in scalable data analytics on Hadoop.*
3. *Develop and manage HBase and Hive databases by creating, manipulating, and querying tables using HiveQL and integrating HBase data storage to solve real-world problems like flight information systems.*
4. *Perform data processing and analysis using Pig Latin scripts to manipulate large datasets with operations like sorting, grouping, joining, and filtering, and manage Hive database objects like tables, views, and indexes.*
5. *Apply techniques for handling diverse data types including JSON, text, CSV, Excel, and XML within Hadoop and R environments, performing data parsing, descriptive statistics, and data subset operations for advanced analytics.*
6. *Implement data visualization in R by creating plots such as boxplots, scatterplots, histograms, bar charts, and pie charts to explore data distributions, detect outliers, and present meaningful insights effectively.*

List of Experiments:

1. Hadoop Installation on a)Single Node and SPARK Installation, Launch a cloud instance for AWS instance on Centos 7.
2. Design a distributed application using MapReduce which processes a log file of a system. List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.
3. Design and develop a distributed application to find the coolest/hottest year from the available weather data. Use weather data from the Internet and process it using MapReduce.
4. Write an application using HBase and HiveQL for flight information system which will include
   a) Creating, Dropping, and altering Database tables,

b) Creating an external Hive table to connect to the HBase for Customer Information Table,
c) Load table with data, insert new values and field in the table, Join tables with Hive,
d) Create index on Flight information Table, and
e) Find the average departure delay per day in 2008.

5. Display the hierarchical structure of your data by generating Trees, graphs and network visualization. Install and Run Pig then write Pig Latin scripts to sort, group, join, project and filter the data. Install and Run Hive then use Hive to Create, alter and drop databases, tables, views, functions and Indexes.
6. Input file contains a series of tweets made by few people. Do a word count on the text object value Hint: Json Parsing in python – this sample snippet can be used within Map to read the JSON
7. Reading different types of data sets (.txt, .csv) from web and disk and writing in file in specific disk location. And Reading Excel,XML data sheets in R. Using with and without R objects on console, mathematical functions on console create R objects for calculator application and save in a specified location in disk.

   Write an R script to find basic descriptive statistics using summary,str, quartile unction on mtcars& cars datasets and to find subset of dataset by using subset (),aggregate () functions on dataset.
8. Implementing data visualization using R : Find the data distributions using box and scatter plot, Find the outliers using plot and Plot the histogram, bar chart and pie chart on sample data.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
-	1	2	2

(25MC9135) DEVOPS & AGILE PROGRAMMING LABORATORY

COURSE OBJECTIVES

1. *To understand the concept of DevOps with associated technologies and methodologies.*
2. *To be familiarized with Jenkins, which is used to build and test software Applications and Continuous integration in DevOps environment.*
3. *To understand different Version Control tools like GIT, CVS or Mercurial*
4. *To understand Docker to build, ship and run containerized images*
5. *To use Docker to deploy and manage Software applications running on Container.*
6. *To be familiarized with concept of Software Configuration Management provisioning using tools like Puppet, Chef, Ansible or Saltstack.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the principles, values, and business benefits of adopting Agile methodologies to improve software development flexibility, collaboration, and customer satisfaction.*
2. *Apply Agile development practices including Test-Driven Development (TDD), design principles, and refactoring techniques to deliver high-quality, maintainable software iteratively.*
3. *Demonstrate competence in using automated build tools and version control systems to manage source code, track changes, and enhance collaboration within Agile teams.*
4. *Configure and utilize continuous integration tools like Jenkins to automate building, testing, and deploying applications, ensuring faster feedback and higher software quality.*
5. *Implement and manage containerization and virtualization technologies such as Docker to package, deploy, and scale applications consistently across diverse environments.*
6. *Set up and operate software configuration management and provisioning tools like Chef, Puppet, Ansible, or Saltstack to automate infrastructure and streamline deployment pipelines in Agile and DevOps environments.*

List of Experiments:

Agile Laboratory Programs:

1. Understand the background and driving forces for taking an Agile Approach to Software Development.
2. Understand the business value of adopting agile approach.
3. Understand agile development practices
4. Drive Development with Unit Test using Test Driven development.
5. Apply Design principle and Refactoring to achieve agility
6. To study automated build tool.
7. To study version control tool.

8. To study Continuous Integration tool.
9. Perform Testing activities within an agile project.

DevOps Laboratory Programs:

1. Build and Test Applications with Continuous Integration - To Install and Configure Jenkins to test, and deploy Java or Web Applications using NetBeans or eclipse.
2. Version Control - To Perform Version Control on websites/Software's using different Version control tools like RCS/ CVS/GIT/Mercurial (Any two)
3. Virtualization and Containerization - To Install and Configure Docker for creating Containers of different Operating System Images
4. Virtualization and Containerization - To Build, deploy and manage web or Java application on Docker
5. Software Configuration Management - To install and configure Software Configuration Management using Chef/Puppet/Ansible or Saltstack.
6. Provisioning - To Perform Software Configuration Management and provisioning using Chef/Puppet/Ansible or Saltstack.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
1	-	2	2

(25MC9137) MEAN STACK DEVELOPMENT
(Skill Oriented Course – II)

COURSE OBJECTIVES

1. *To understand basic concepts of JAVASCRIPT.*
2. *To implement concepts of HTML, CSS, and REACT in developing various websites.*
3. *To design solutions to real world scenarios using NODE and EXPRESS JS.*
4. *To Analyze concepts of MONGODB.*
5. *To implement socket programming in MERN stack.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand and utilize core JavaScript concepts including data types, functions, object-oriented programming, asynchronous programming with promises and async/await, modules, error handling, and DOM manipulation using jQuery.*
2. *Design and create responsive web pages using HTML and CSS, and develop interactive user interfaces with React, leveraging virtual DOM, JSX, component lifecycle, props, and state management.*
3. *Develop server-side applications using Node.js and Express.js, handling HTTP requests and responses, routing, middleware, file uploads, cookies, and asynchronous event-driven programming.*
4. *Work with MongoDB for NoSQL database management, including installation, creating and dropping databases and collections, inserting and querying documents, data modeling, and sorting data.*
5. *Implement real-time communication in MERN applications using socket programming and integrate React front-end with Node.js server and MongoDB database via Axios for full stack development.*
6. *Build and deploy complete full-stack MERN projects by connecting React front-end components with Node.js/Express back-end services and MongoDB data storage, demonstrating end-to-end application development skills.*

UNIT – I

Introduction: data types, logical operations, functions, object and classes, promise async and await, modules and npm packages, error handling, Document Object module, J Query.

UNIT – II

HTML CSS and REACT: Basic structure of a webpage, Different types of tags, HTML text fundamentals, creating hyperlinks, Insertion of images and multimedia, Introduction CSS, CSS selector –internal- external, CSS- inline class background font text color, CSS-padding margin border, Installation of react, REACT- virtual DOM, REACT-JSX, REACT-components, REACT-prop and state, REACT – lifecycles.

UNIT – III

Node and Express JS: Introduction of Node JS (Run time environment), Node JS installation, Node JS web based example (import required modules ,create server, read request and return response), Node JS – npm ,errors, crypto, Node JS – child process ,buffer, string, Node JS- string decoder ,query string , Node JS- callbacks , events, web modules, Introduction of APIs, Express JS – introduction , Express JS- installation, Express JS – GET, POST, REQUEST, RESPONSE, Express JS- Routing ,file upload, cookies, middleware.

UNIT – IV

MongoDB: Introduction of MongoDB, Difference between SQL and NoSQL, MongoDB data types, MongoDB installation, Data modelling in MongoDB, create database, Drop Database, create collection, insert document, select document, Queries in MongoDB, Sorting data in document, Remove document.

UNIT – V

Socket programming in MERN stack: Connect the react to node by axiom, Import required module, create server in node, Connect the Node JS to MongoDB, create request, Read Response, Full Stack Project.

TEXT BOOKS

1. Getting MEAN with MONGO, Express angular and node by Simon Holmes, Dreamtech Publishers
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node Paperback – 1 April 2017 by Vasan Subramanian (Author)
3. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App by Greg Lim (Author)
4. Full Stack JavaScript Development with MEAN by COLIN J Ihrig and Adam J bretz.Sitepoint publishers.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(25MC9138) DEEP LEARNING
(Programme Elective – IV)

COURSE OBJECTIVES

1. *To present the mathematical, statistical and computational challenges of building neural networks.*
2. *To teach the concepts of deep learning.*
3. *To introduce dimensionality reduction techniques.*
4. *To enable the students to know deep learning techniques to support real-time applications.*
5. *To explain the case studies of deep learning techniques.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Apply fundamental machine learning algorithms, including linear models, neural networks, and logistic regression, to solve classification and regression problems.*
2. *Analyze and design deep neural networks using concepts like backpropagation, regularization, batch normalization, and probabilistic deep learning theories.*
3. *Implement convolutional neural networks (CNNs) and advanced generative models (GANs) for supervised, semi-supervised, and unsupervised learning tasks.*
4. *Employ dimensionality reduction techniques (PCA, LDA, autoencoders) and optimize ConvNet architectures (AlexNet, VGG, Inception, ResNet) for high-performance learning.*
5. *Generalize deep learning models using stochastic and non-convex optimization methods, recurrent neural networks, and reinforcement learning strategies.*
6. *Develop real-world AI applications in domains such as image recognition, nlp, bioinformatics, and audio processing through case studies.*

UNIT – I

Introduction: Introduction to machine learning- Linear models (SVMs and Perceptron, logistic regression)

Intro to Neural Nets: What a shallow network computes- Training a network: loss functions, back propagation and stochastic gradient descent- Neural networks as universal function approximates.

UNIT - II

Deep Networks: History of Deep Learning- A Probabilistic Theory of Deep Learning- Back propagation and regularization, batch normalization- VC Dimension and Neural Nets-Deep Vs Shallow Networks

Convolutional Networks - Generative Adversarial Networks (GAN), Semi- supervised Learning.

UNIT - III

Dimensionality Reduction: Linear (PCA, LDA) and manifolds, metric learning - Auto encoders and dimensionality reduction in networks - Introduction to Convnet - Architectures – AlexNet, VGG, Inception, ResNet - Training a Convnet: weights initialization, batch normalization, hyper parameter optimization.

UNIT - IV

Optimization and Generalization: Optimization in deep learning– Non-convex optimization for deep networks- Stochastic Optimization Generalization in neural networks- Spatial Transformer Networks- Recurrent networks, LSTM - Recurrent Neural Network Language Models- Word-Level RNNs & Deep Reinforcement Learning - Computational & Artificial Neuroscience.

UNIT - V

Case Study and Applications: Image net- Detection-Audio Wave Net-Natural Language Processing Word2Vec - Joint Detection Bioinformatics- Face Recognition- Scene Understanding- Gathering Image Captions.

TEXT BOOKS

1. Deep Learning", Ian Goodfellow, YoshuaBengio , Aaron Courville, MIT Press 2016.
2. "Neural Networks and Deep Learning A Text Book", Charu C Aggarwal, Springer International Publishing AG, Part of Springer Nature 2018.

REFERENCES

1. Josh Patterson, Adam Gibson "Deep Learning: A Practitioner's Approach", O'Reilly Media, 2017
2. 2. Umberto Michelucci "Applied Deep Learning. A Case-based Approach to Understanding Deep Neural Networks" Apress, 2018.
3. 3. Kevin P. Murphy "Machine Learning: A Probabilistic Perspective", The MIT Press, 2012.
4. 4. Ethem Alpaydin,"Introduction to Machine Learning", MIT Press, Prentice Hall of India, Third Edition 2014.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(25MC9139) SOCIAL MEDIA ANALYSIS
(Programme Elective – IV)

COURSE OBJECTIVES

1. *To inspire the students with interest, excitement, and urge to learn the subject of Social network analysis.*
2. *To understand the fundamental concepts of Social network analysis .*
3. *To introduce the purpose of learning important aspects in Social network analysis.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Explain the concepts of the Social Web, Semantic Web, and Social Network Analysis, including their evolution, key measures, and practical applications.*
2. *Apply ontology-based knowledge representation techniques, including RDF and OWL, to model and aggregate social network data for semantic web applications.*
3. *Analyze community structures in social networks using community detection and mining algorithms, and evaluate their effectiveness in real-world applications.*
4. *Predict user behavior in social networks using data inference, context-awareness, and reality mining, while considering privacy implications.*
5. *Evaluate trust models, trust transitivity, and reputation systems in online communities, and propose strategies for enhancing trust and security.*
6. *Develop solutions for predicting, and securing online social networks by integrating network analysis, semantic representation, and trust evaluation techniques.*

UNIT – I

Introduction To The Social Web And Network Analysis: Introduction to Web - Limitations of current Web – Development of Semantic Web – Emergence of the Social Web - Network analysis -Development of Social Network Analysis - Key concepts and measures in network analysis - Electronic sources for network analysis -Electronic discussion networks, Blogs and online communities, Web-based networks - Applications of Social Network Analysis

UNIT - II

Semantic Web And Ontological Representations: Ontology and their role in the Semantic Web - Ontology-based Knowledge Representation - Ontology languages for the Semantic Web -RDF and OWL - Modeling and aggregating social network data – State-of-the-art in network data representation, Ontological representation of social individuals - Ontological representation of social relationships, Aggregating and reasoning with social network data, Advanced Representations

UNIT - III

Community Detection And Web Archive Analysis: Extracting evolution of Web Community from a Series of Web Archive - Detecting Communities in Social Networks - Definition of Community - Evaluating Communities -–Methods for Community Detection &

Mining -Applications of Community Mining Algorithms- Tools for Detecting Communities Social Network Infrastructures and Communities-Applications - Case Studies- Real Time Sentiment Analysis, Stock Market Predictions.

UNIT - IV

Behavior Prediction And Privacy In Social Networks: Understanding and Predicting Human Behavior for Social Communities - User Data Management, Inference and Distribution- Enabling New Human Experiences - Reality Mining - Context-Awareness - Privacy in Online Social Networks

UNIT - V

Trust And Security In Online Communities: Trust in Online Environment - Trust Models Based on Subjective Logic - Trust Network Analysis - Trust Transitivity Analysis - Combining Trust and Reputation - Trust Derivation Based on Trust Comparisons - Attack Spectrum and Countermeasures

TEXT BOOKS

1. Charu C. Aggarwal, "Social Network Data Analytics", Springer, 2011.
2. GuandongXu ,Yanchun Zhang and Lin Li, "Web Mining and Social Networking Techniques and applications", Springer, first edition, 2011.

REFERENCES

1. Peter Mika, "Social networks and the Semantic Web", Springer, first edition 2007.
2. BorkoFurht, "Handbook of Social Network Technologies and Applications", Springer, first edition, 2010.
3. Dion Goh and Schubert Foo, "Social information retrieval systems: emerging technologies and applications for searching the Web effectively", IGI Global snippet, 2008. 133
4. Max Chevalier, Christine Julien and Chantal Soulé-Dupuy, "Collaborative and social information retrieval and access: techniques for improved user modelling", IGI Global snippet, 2004 .

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(25MC9140) MULTIMEDIA SYSTEMS AND TOOLS
(Programme Elective – IV)

COURSE OBJECTIVES

1. *Formulate a working definition of interactive multimedia*
2. *Demonstrate competence in using the authoring program Hyper Studio*
3. *Outline the use of animation, digitized sound, video control, and scanned images*
4. *Illustrate the use of Netscape to access the Course Home Page and Tips and Tricks*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand multimedia concepts, applications, and authoring tools, including HyperStudio, Adobe Premiere, Macromedia Director, Flash, and Dreamweaver.*
2. *Apply instructional and screen design principles to develop engaging multimedia content with effective layouts, navigation, and visual consistency.*
3. *Use multimedia elements such as sound, graphics, animation, and web documents to create interactive applications.*
4. *Design and develop multimedia portfolios with advanced interaction features, including hyperlinks, drag-and-drop, and custom actions.*
5. *Demonstrate knowledge on various digital media formats, including QuickTime movies, Laserdisc, CD-ROM, and scanned content, within multimedia projects.*
6. *Develop complete multimedia applications by combining authoring tools, instructional design strategies, and digital media integration techniques.*

UNIT – I

Introduction To Multimedia And Authoring Tools: Multimedia Overview, Definition Applications and Design, Authoring (HyperStudio), Introduction to HyperStudio, The Metaphor, The Basics (Cards, Buttons, Text), HyperStudio, Resources. Multimedia Authoring - Multimedia Authoring Metaphors, Multimedia Production, Multimedia Presentation, Automatic Authoring, Some Useful Editing and Authoring Tools, Adobe Premiere, Macromedia Director, Macromedia Flash, Dreamweaver.

UNIT - II

Instructional And Screen Design In Multimedia: Instructional Design, Objectives, Content (print, graphics, sounds, etc.), Interaction, Assessment, Closure, Screen Design: Metaphors and Themes, Colors and Backgrounds, Text (size, color, placement), Navigation, Consistency.

UNIT - III

Multimedia Interactivity And Integration Techniques: Transitions and Links, Use of Sound, HyperStudio Sounds, Recording Your Own, Internet Resources, Graphics, Integrating Web documents, HyperStudio Tips and Tricks, Animation, Launching other applications and documents

UNIT - IV

Creating Multimedia Portfolios And Advanced Interactions: Multimedia Portfolios, Designing a template, Adding elements, Choosing materials, Advanced Button Features, Hyperlinks, Drag-n-Drop, Advanced NBA's, Using Actions with other Objects.

UNIT - V

Digital Media Integration And Control Mechanisms: Incorporating Digital Media, QuickTime Movies, Laserdisc and CD-ROM control, scanning.

TEXT BOOKS

1. Marcia Kuperberg, A Guide to Computer Animation: for TV, games, multimedia and web,Focal Press (Taylor and Francis Group), 2002.
2. Z. N. Li and M. S. Drew, “Fundamentals of Multimedia”, Pearson Prentice HallKenneth H Rosen, 7thEdition, TMH.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(25MC9141) CYBER LAWS
(Open Elective – II)

COURSE OBJECTIVES

1. *The objectives of this course are to enable the learner to understand, explore, and acquire a critical understanding of cyber laws.*
2. *Equip the learner with competencies for dealing with frauds and deceptions, and other cybercrimes that take place via the Internet*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand the conceptual and theoretical foundations of cyber law, its national and international development, and its role in ensuring cyber security and data protection.*
2. *Analyze jurisdictional challenges in cybercrime, including privacy concerns, hacking incidents, and global conventions such as the Budapest Convention.*
3. *Apply cyber law principles to intellectual property rights in the digital domain, including copyright, trademarks, domain names, and database protection.*
4. *Evaluate the legal aspects of e-commerce, including online transactions, e-banking, e-taxation, and contractual obligations in digital environments.*
5. *Interpret the provisions of the Information Technology Act 2000 and related legal frameworks, including electronic governance, digital signatures, and the role of certifying authorities.*
6. *Assess legal issues related to internet service providers, social networking sites, and emerging trends in cyber regulations at the national and international level.*

UNIT – I

Foundations Of Cyber Law And Cyber Security: Conceptual and theoretical perspective of Cyber Law, Computer and Web Technology, Development of Cyber Law, National and International Perspective Cyber Law, Legal issues and challenges in India, USA, Data Protection, Cyber Security.

UNIT - II

Jurisdiction And Privacy In Cyber Crime: Jurisdiction issues in Transactional Crimes Cyber Law, International Perspective, Budapest Convention on Cybercrime. Hacking and Legal Issues, Privacy legal issues

UNIT - III

Cyber Law And Intellectual Property Rights(IPR): Cyber Law and IPR, Understanding Copyright in Information Technology, Software Copyrights Copyright in Internet & Multimedia, Software Piracy, Trademarks in Internet Domain Name registration, Domain Name disputes, Icann's core principles and domain names, Net Neutrality, Databases in IT, Protection of databases, Position in USA, EU and India.

UNIT - IV

Legal Aspects Of E-Commerce: E-Commerce, UNCITRAL Model, Legal Aspects of E-Commerce, E-Taxation, E-Banking, Online Publishing and online credit card payment, Employment Contracts, Non-Disclosure Agreements.

UNIT - V

Information Technology Act And Legal Frameworks: Information Technology Act 2000, Aims and Objectives, Overview of the Act, Jurisdiction, Electronic Governance, Electronic Evidence, Digital Signature Certificates, Digital Signatures, Duties of Subscribers, Role of Certifying Authorities, Regulations Appellate Tribunal, Internet Service Providers and their liabilities, Social Networking Sites.

TEXT BOOKS

1. Law Relating to Computer, Internet and E-Commerce by KamathNandan, 5thEdition,Universal Law Publishing.

REFERENCES

1. Marjie T. Britz - Computer Forensics and Cyber Crime: An Introduction – Pearson.
2. Chwan-Hwa (John) Wu,J. David Irwin - Introduction to Computer Networks and Cyber security Press.
3. Bill Nelson, Amelia Phillips, Christopher Steuart - Guide to Computer Forensics and Investigations Cengage Learning.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(25MC9142) ENTREPRENEURSHIP
(Open Elective – II)

COURSE OBJECTIVES

1. *The aim of this course is to have a comprehensive perspective of inclusive learning, ability to learn and implement the fundamentals of Entrepreneurship.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand concepts, types, and competencies of entrepreneurship, along with the processes and models for entrepreneurial development.*
2. *Evaluate business opportunities and prepare effective business plans, considering institutional support systems at central and state levels.*
3. *Analyze the challenges faced by MSMEs, identify causes of industrial sickness, and propose rehabilitation strategies for sick enterprises.*
4. *Apply marketing strategies, pricing techniques, branding, and international trade practices to promote and grow enterprises.*
5. *Formulate strategic growth plans for entrepreneurial ventures, including valuation, harvesting strategies, and technology adoption.*
6. *Assess the role of women entrepreneurs, and identify strategies, policies, and institutional support mechanisms for their development in India.*

UNIT – I

Entrepreneurial Perspectives: Introduction to Entrepreneurship – Evolution - Concept of Entrepreneurship - Types of Entrepreneurs - Entrepreneurial Competencies, Capacity Building for Entrepreneurs.

Entrepreneurial Training Methods- Entrepreneurial Motivations - Models for Entrepreneurial Development - The process of Entrepreneurial Development.

UNIT - II

New Venture Creation: Introduction, Mobility of Entrepreneurs, Models for Opportunity Evaluation; Business plans – Purpose, Contents, Presenting Business Plan, Procedure for setting up Enterprises, Central level - Startup and State level - T Hub, Other Institutions initiatives.

UNIT - III

Management of MSMEs and Sick Enterprises: Challenges of MSME s, Preventing Sickness in Enterprises – Specific Management Problems; Industrial Sickness; Industrial Sickness in India – Symptoms, process and Rehabilitation of Sick Units

UNIT - IV

Managing Marketing and Growth of Enterprises: Essential Marketing Mix of Services, Key Success Factors in Service Marketing, Cost and Pricing, Branding, New Techniques in Marketing, International Trade.

UNIT - V

Strategic perspectives in Entrepreneurship: Strategic Growth in Entrepreneurship, The Valuation Challenge in Entrepreneurship, The Final Harvest of New Ventures, Technology, Business Incubation, India way – Entrepreneurship; Women Entrepreneurs – Strategies to develop Women Entrepreneurs, Institutions supporting Women Entrepreneurship in India.

TEXT BOOKS

1. Entrepreneurship Development and Small Business Enterprises, Poornima M.Charantimath, 2nd edition, Pearson, 2014.
2. Entrepreneurship, a South – Asian Perspective, D.F.Kuratko and T.V.Rao, 3rd edition, Cengage, 2012.
3. Entrepreneurship, Arya Kumar, 4th edition, Pearson 2015.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(25MC9143) NOSQL DATABASES
(Open Elective – II)

COURSE OBJECTIVES

1. *Distinguish the different types of NoSQL databases. Understand the impact of the cluster on database design. State the CAP theorem and explain it main points.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Understand types of NoSQL databases—document-oriented, key-value, column-oriented, and graph—along with their architectures, features, and use cases.*
2. *Apply data modeling, loading, querying, and performance tuning techniques for document-oriented and column-oriented NoSQL databases.*
3. *Analyze the relationship between relational databases with NoSQL systems, and evaluate the suitability of different NoSQL models for specific applications.*
4. *Implement replication, sharding, and MapReduce techniques to enable scalability, fault tolerance, and distributed processing in NoSQL databases.*
5. *Design and manage applications using MongoDB, Cassandra, HBase, and Riak by selecting appropriate consistency, availability, and query models for real-world scenarios.*
6. *Assess the limitations and challenges of different NoSQL models, identifying scenarios where they are unsuitable, and propose alternative data storage solutions.*

UNIT – I

Define, compare and use the four types of NoSQL Databases (Document-oriented, KeyValue Pairs, Column-oriented and Graph). Demonstrate an understanding of the detailed architecture, define objects, load data, query data and performance tune Column-oriented NoSQL databases. Explain the detailed architecture, define objects, load data, query data and performance tune Document- oriented NoSQL databases.

UNIT - II

Comparison of relational databases to new NoSQL stores, MongoDB, Cassandra, HBASE, Neo4j use and deployment, Application, RDBMS approach, Challenges NoSQL approach, Key-Value and Document Data Models, Column-Family Stores, Aggregate-Oriented Databases.

UNIT - III

Replication and sharding, MapReduce on databases. Distribution Models, Single Server, Sharding, Master- Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication.NoSQL Key/Value databases using MongoDB, Document Databases, What Is a Document Database? Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Event Logging, Content Management Systems, Blogging Platforms, Web Analytics or Real-Time Analytics, E-Commerce Applications, When Not to

Use, Complex Transactions Spanning Different Operations, Queries against Varying Aggregate Structure.

UNIT - IV

Column- oriented NoSQL databases using Apache HBASE, Column-oriented NoSQL databases using Apache Cassandra, Architecture of HBASE, What Is a Column-Family Data Store? Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Event Logging, Content Management Systems, Blogging Platforms, Counters, Expiring Usage, When Not to Use.

UNIT - V

NoSQL Key/Value databases using Riak, Key-Value Databases, What Is a Key-Value Store, Key-Value Store Features, Consistency, Transactions, Query Features, Structure of Data, Scaling, Suitable Use Cases, Storing Session Information, User Profiles, Preferences, Shopping Cart Data, When Not to Use, Relationships among Data, Multioperation Transactions, Query by Data, Operations by Sets.

TEXT BOOKS

1. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence Sadalage, P. &FowlerPearson Education