

**Department of Computer Applications****2 Year MCA Entry – Course Structure (with effective from the academic year 2014 - 2015)**

Semester	Course Number	Course Title	Instru. Periods per week				Credits	Max. Marks		Total
			L	T	P	Total		Sessional	University	
First	14IMCA21T01	Advanced Database Management System	4			4	4	40	60	100
	14IMCA21T02	Design and Analysis of Algorithm	4			4	4	40	60	100
	14IMCA21T03	Computer networks	4			4	4	40	60	100
	14IMCA21T04	Accountancy and Financial Management	4			4	4	40	60	100
	14IMCA21T05	Programming to python	4			4	4	40	60	100
	14IMCA21P01	Design and analysis of algorithms Lab			3	3	3	40	60	100
	14IMCA21P02	Database Management Systems Lab			3	3	3	40	60	100
	14IMCA21P03	Python Programming –Lab			3	3	3	40	60	100
						<b>Total</b>	<b>29</b>		<b>Total</b>	<b>700</b>
	Second	14IMCA22T06	Software Engineering	4			4	4	40	60
14IMCA22T07		Web Programming through PHP	4			4	4	40	60	100
14IMCA22T08		Operating Systems	4			4	4	40	60	100
14IMCA22T09		Network Security essentials applications and standards	4			4	4	40	60	100
14IMCA22T10		Computer Organization	4			4	4	40	60	100
14IMCA22P04		Web Programming-Lab-PHP			3	3	3	40	60	100
14IMCA22P05		Operating Systems Lab			3	3	3			
14IMCA22P06		Network Security Lab			3	3	3	40	60	100
						<b>Total</b>	<b>29</b>		<b>Total</b>	<b>700</b>
Third		14IMCA31T11	Data Mining and Warehouse.	4			4	4	40	60
	14IMCA31T12	Mobile Application Development using Android.	4			4	4	40	60	100
	14IMCA31T13	Information Retrieval System	4			4	4	40	60	100
	14IMCA31T14	Elective –II 51E04-A: Cloud Infrastructure 51E04-B: Decision Support Systems , 51E04-C: Enterprise Resource planning	4			4	4	40	60	100
	14IMCA31T15	Object oriented analysis&design	4							
	14IMCA31P07	Unified Modelling Language Lab			3	3	3	40	60	100
	14IMCA31P08	Data Warehouse and Mining Lab			3	3	3	40	60	100
	14IMCA31P09	Mobile Application Development Lab Android			3	3	3	40	60	100
						<b>Total</b>	<b>29</b>		<b>Total</b>	<b>700</b>
	Fourth	14IMCA32T16	Big Data and Analytics	4			4	3	40	60
14IMCA32T17		Software Project Management	4			4	3	40	60	100
14IMCA32D01		Project Work /Seminars.					18			
						<b>Total</b>	<b>24</b>			
						<b>111</b>				

L- Lecture

T-Tutorial

P-Practical

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

2 Year MCA - I SEM

L	T	C
4	0	4

**ADVANCED DATABASE MANAGEMENT SYSTEM**

**(14MCA21T01)**

**Course Objectives:**

- *To understand the role of a database management system in an organization.*
- *To understand basic database concepts, including the structure and operation of the relational data model.*
- *To construct simple and moderately advanced database queries using Structured Query Language (SQL).*
- *To Understand and successfully apply logical database design.*
- *Principles, including E-R diagrams and database normalization.*
- *To Design and implement a small database project using Microsoft Access.*
- *To understand the concept of a database transaction and related database facilities, including concurrency control, journaling, backup and recovery, and data object locking and protocols.*
- *To Describe and discuss selected advanced database topics, such as distributed database systems and the data warehouse.*
- *To understand the role of the database administrator.*

**Course Outcomes:**

- *Upon successful completion of this course, students should be able to:*
- *Describe fundamental elements of a relational database management system.*
- *Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and database language SQL.*
- *Identify other data models such as object-oriented model and XML model Skill*
- *Design entity-relationship diagrams to represent simple database application scenarios.*
- *Convert entity-relationship diagrams into relational tables, populate a relational database and formulate SQL queries on the data.*
- *Criticize a database design and improve the design by normalization.*

**UNIT- I**

**OVERVIEW -THE RELATIONAL MODEL OF DATA:** Theoretical concepts Relational model conformity and Integrity, Relational Database Management Systems Implementation, Techniques, file system, storage management.

**PL/SQL** – Introduction to PL/SQL, Declare, begin statements, Variables, Control Structure.

**PL/SQL TRANSACTIONS** – Save point, Cursor, PL/SQL Database Objects – Procedures, Functions, Packages, Triggers.

**PROGRAMMATIC SQL** – Embedded SQL, Dynamic SQL, and ODBC Standard.

**UNIT-II**

**TRANSACTION PROCESSING AND CONCURRENCY CONTROL:** Definition of Transaction and ACID properties. **TRANSACTION PROCESSING** - Transaction-processing monitors, transactional workflows, main-memory databases, real-time transaction systems, long-duration transactions, transaction management in multi-databases. **CONCURRENCY CONTROL** – Locks, Optimistic Concurrency Control (Backward and Forward validations), Time stamping Concurrency Control.

#### **UNIT -III**

**OBJECT-BASED DATABASES AND XML:** Object-based databases – Complex data types, structured types and inheritance in SQL, table inheritance, array and multi set types in SQL, object identity and reference types in SQL, implementing O-R features, Persistent programming languages, OO vs OR. **XML:** Structure of XML, Document Schema, Querying and Transformation, API in XML, XML applications.

#### **UNIT -VI**

**DATABASE SECURITY:** Security and integrity threats, Defense mechanisms, Statistical database auditing & control. Security issue based on granting/revoking of privileges, Introduction to statistical database security. **PL/SQL SECURITY:** Locks ,Implicit locking, types and levels of locks, explicit locking, Oracles' named Exception Handlers.

#### **UNIT- V**

**EMERGING DATABASE MANAGEMENT SYSTEM TECHNOLOGIES:** Object oriented, deductive, spatial, and temporal and constraint database management Systems, New database applications and environments: e.g. Data Warehousing; Multimedia; Mobility; Multidatabases; Native XML databases (NXD), Internet DATABASE RELATED STANDARDS SQL standards, SQL 1999, SQL:2003, Object Data Management Group (ODMG) version 3.0 standard, Standards for interoperability and integration e.g. Web Services, SOAP XML related specifications, e.g. XQuery, XPath.

#### **Textbooks:**

1. Database Management Systems Raghu Ramkrishnan, Gehrke
2. Database Management Systems – Abraham Silbertz, S.Sudershan, Henry Korth

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

2 Year MCA - I SEM

L	T	C
4	0	4

**DESIGN AND ANALYSIS OF ALGORITHMS**

**(14MCA21T02)**

**Course Objectives:**

*Upon completion of this course, students will be able to do the following:*

- *Analyze the asymptotic performance of algorithms.*
- *Write rigorous correctness proofs for algorithms.*
- *Demonstrate a familiarity with major algorithms and data structures.*
- *Apply important algorithmic design paradigms and methods of analysis.*
- *Synthesize efficient algorithms in common engineering design situations.*

**Course Outcomes:**

*Students who complete the course will have demonstrated the ability to do the following:*

- *Argue the correctness of algorithms using inductive proofs and invariants.*
- *Analyze worst-case running times of algorithms using asymptotic analysis.*
- *Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.*
- *Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.*
- *Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.*
- *Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.*
- *Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Recite algorithms that employ randomization. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.*
- *Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.*

**UNIT-I**

**INTRODUCTION:** Algorithm Specification, Performance analysis, Space Complexity, Time Complexity, Asymptotic Notation(O, Omega, Theta), Practical Complexities, Performance Measurement, Review of elementary data structure- Heap and Heap Sort, Hashing, Set representation. UNION, FIND.

## UNIT-II

**DIVIDE-AND CONQUER:** The general method, finding maximum, minimum. Merge sort quick sort and selection. Greedy Method: Knapsack problem, Optimal Storage on tapes, Job sequencing with deadlines, Optimal merges patterns, Minimum Spanning Trees.

## UNIT-III

**DYNAMIC PROGRAMMING AND TRAVERSAL TECHNIQUE:** Multistage graph, All Pair Shortest Path, Optimal Binary Search trees, 0/1 Knapsack, Reliability Design, Traveling Salesman Problem, Bi connected Components and Depth First Search.

## UNIT-IV

**BACKTRACKING AND BRANCH AND BOUNDS:** 8-Queens Problem, Graph Coloring Hamilton cycle, Knapsack Problem, 0/1 Knapsack Problem, Traveling salesperson problem, Lower-Bound Theory.

## UNIT-V

**NP-HARD AND NP-COMPLETENESS:** Basic concepts, cook's theorem, NP hard graph problems and scheduling problem, NP-hard code generation problems, Clique Decision problem, Node covering decision, Scheduling problem, NP hard code generation problem.

### Text Books:

1. Horowitz E. Sahani S: "Fundamentals of Computer Algorithm", Galgotia Publications.
2. Anany Levitin, "Introduction to the Design & Analysis, of Algorithms", Pearson Education, 2000.
3. Aho, Hopcroft, Ulman, "The Design and Analysis of Computer Algorithm", Pearson Education, 2000.
4. Parag H. Dave, Himanshu B. Dave "Design and Analysis of Algorithms" Pearson Education, 2008.

2 Year MCA - I SEM

L	T	C
4	0	4

**COMPUTER NETWORKS  
(14MCA21T03)**

**Course Objectives**

At the end of the course, the students will be able to:

- Build an understanding of the fundamental concepts of computer networking, modern network architectures and its components from a design and performance perspective.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the students for entering into advanced courses in computer networking and creating an opportunity to do network programming.
- Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

**Course Outcomes**

After completing this course the student must demonstrate the knowledge and ability to:

- Independently understand basic computer network technology and they can explain Data Communications System and its components.
- Identify the different types of network topologies, protocols and layers of the OSI model and TCP/IP. Students will be able to explain the function(s) of each layer.
- Identify the different types of network devices and their functions within a network
- Understand the concept of sub-netting and routing mechanisms.
- Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

**UNIT-I**

**INTRODUCTION:** Datacommunications, Networks, The Internet, and Protocol & Standards Network Models: layered tasks, Internet model, OSI model, TCP model.

**UNIT-II**

**PHYSICAL LAYER:** Signals: Analog and digital signals, data rate limits, Transmission impairment, Signal measurements like throughput, propagation speed and time, wave length Digital Transmission: Line coding, block coding, sampling, transmission mode Analog Transmission: Modulation digital data, telephonemodem, Modulation analog signals Multiplexing: FDM, WDM, TDM-Transmission Media: Guided media, unguided media Circuit Switching & Telephone Network: Circuit switching, telephone network.

**UNIT-III**

**DATALINK LAYER:**

Error detection and Correction: Type of errors, detection and correction of errors Data Link Control & Protocol: Flow & error control, Stop-And-Wait ARQ, Go-Back-NARQ, Select Repeat ARQ, HDLC-Point-To-Point Access: Point-to-point protocol, PPP stack Local Area Network: Traditional Ethernet, fast and gigabit Ethernets Connecting LANs, Backbone Networks and Virtual LANs: Connecting devices, Backbone networks, Virtual LANs.

#### **UNIT-IV**

**NETWORK LAYER:** Internetworks, Addressing, Routing - Network Layer Protocols: ARP, IP, ICMP, IPV6 – Unicast routing, Unicast routing protocols, Multirouting, Multicast routing protocols

#### **UNIT-V**

**TRANSPORT LAYER:** Process-To-Process delivery, user datagram, Transmission control protocol.

**APPLICATION LAYER:** Client-Server Model: Client- Server model, Socket interface – A brief introduction to DNS, SMTP, FTP

#### **Text Books:**

- 1) Data Communications and Networking, Behrouz A. Forouzan, 4th Edition, Tata Mcgraw-Hill Publishing Co

#### **Reference Book:**

- 1) Understanding Data Communications and Networks, William A. Shay, 2nd Edition, Vikas Publishing House

2 Year MCA - I SEM

L	T	C
4	0	4

**ACCOUNTING AND FINANCIAL MANAGEMENT  
(14IMCA21T04)**

**Course Objectives:**

- *The Accountancy and financial Management subject encourages students to learn about accounting principles, various types of accounting, cost accounting, budgeting, various ratios.*
- *The Accountancy and financial Management subject prepares students to continue into graduate education.*
- *The Accountancy and financial Management subject prepares students to enter into accounting positions in private, public or governmental organizations.*

**Course Outcomes:**

*Upon completion of Accountancy and financial Management subject students will be able to:*

- *Demonstrate an understanding of the functional areas of accounting, marketing, finance, management, and economics.*
- *Demonstrate an understanding of the legal and social environment of business.*
- *Demonstrate an understanding of the global environment of business.*
- *Demonstrate an understanding of the ethical obligations and responsibilities of business.*
- *Demonstrate the ability to use business tools.*
- *Demonstrate the ability to communicate effectively.*
- *Demonstrate the ability to apply knowledge of business concepts and functions in an integrated manner.*

**UNIT -I**

**INTRODUCTION TO ACCOUNTING:** Principles, concepts and conventions, double entry system of accounting, classification of accounts and debit-credit rules.

**Financial Statements:** Introduction to basic books of accounts, journal and ledger – trial balance - preparation of final accounts: trading account, profit and loss account and balance sheet.

**UNIT -II**

**INTRODUCTION TO FINANCIAL MANAGEMENT:** Meaning and scope, role of financial manager, objectives of time value of money ,goals of financial management.

**LEVERAGES:** operation, financial leverage and combined leverage.

**CAPITAL STRUCTURE:** Cost of capital: cost of equity, preference shares, bonds – weighted average cost of capital – capital gearing – overcapitalization and undercapitalization, sources of finance.

**UNIT -III**

**FINANCIAL ANALYSIS THROUGH RATIOS:** Ratio Analysis – classification of ratios – short term solvency and

long term solvency – profitability ratios – analysis and interpretation of financial statements through ratios of liquidity, solvency and profitability.

**FUNDS FLOW AND CASH FLOW ANALYSIS:** Meaning, Importance, statement of changes in working capital, statement of sources and application of funds.

**CASH FLOW ANALYSIS:** cash flow statements, preparation, analysis and interpretation.

#### **UNIT -IV**

**BREAK EVEN ANALYSIS:** Concept of Break Even Point, Cost-Volume-Profit Analysis, Determination of Break Even Point, Margin of Safety and P/V ratio, Impact of changes in cost or selling price on BEP, Practical applications of Break Even Analysis.

#### **UNIT -V**

**CAPITAL BUDGETING:** Capital and its significance, types of capital, estimation of fixed and working capital requirements, methods and sources of raising capital.

**CAPITAL BUDGETING:** features, proposals, methods of capital budgeting, payback method, accounting rate of return (AAR), Net Present Value Method(NPV) and Internal Rate of Return (IRR) -simple problems.

#### **Text Books:**

- 1) Financial Accounting, S.N.Maheshwari, Sultan Chand, 2009.
- 2) Financial Management and Policy, Van Horne, James,C., Pearson ,2009.
- 3) Financial Accounting, Tulsian, S Chand, 2009.
- 4) Financial Statement Analysis, Khan and Jain, PHI, 2009
- 5) Financial Management, I.M.Pandey, Vikas Publications
- 6) Financial Management, Bhat Sundhindra, Excel: 2009
- 7) Financial Management, Prasanna Chandra, T.M.H, 2009.

2 Year MCA - I SEM

L	T	C
4	0	4

**PROGRAMMING TO PYTHON**

**(14MCA21T05)**

**Course Objectives:**

- To develop skills to design and analyze the problems.
- Students should be able to design a flowchart for any real time problem.
- To inculcate the programming skills of python language.
- To make the students to feel comfort with the object oriented programming.

**Course Outcomes:**

- Students will able to design flowcharts for any specific problem.
- Students can learn python language constructs.
- Students will learn OOP concepts which will be helpful for them in further semesters.
- Students will have basic idea on data structures.

**UNIT -I**

**INTRODUCTION TO COMPUTERS:** Introduction to computers, Computer definition, Block diagram of Computer, Hardware Vs Software, Software development life cycle, Structured programming, Computer languages, Creating and running the programs, Number Systems. Introduction to Computer problem solving: Introduction, The problem solving aspect, Top down design, Bottom-up Approach, Implementation of algorithms, The efficiency of Algorithms, Basic Computing Steps and Flow charting (Assignment, Sequencing, Conditionals, Iterations) Practical: Scratch, Raptor

**UNIT -II**

**VARIABLES, EXPRESSIONS AND STATEMENTS:** Values and types, keywords, Operators Expressions, Interactive mode and script mode, String operations, Comments. Functions & Modules: Function calls, Type conversion functions, Math functions, Adding new functions, Definitions and uses, Flow of execution, Parameters and arguments, Random numbers, The time module, The math module. Conditionals: Conditional execution, Alternative execution, Chained conditionals, Nested conditionals. Iteration: Multiple assignment, Updating variables, the while statement, break, continue.

**UNIT -III**

**STRINGS:** A string is a sequence, Traversal with for loop, String slices, Strings are immutable, Searching, Looping and counting, String methods, the in operator, String comparison. Tuples: Tuples are immutable, Tuple assignment, Tuples as return values, Lists and tuples, Dictionaries and tuples, Comparing tuples, Sequences of sequences, Debugging. Lists: Traversing a list, List operations, List

slices, List methods. Recursion: Stack diagrams for recursive functions, Infinite. Files: Persistence, Reading and writing, Filenames and paths

#### **UNIT -IV**

**CLASSES AND OBJECTS:** User-defined types, Attributes, Instances as return values Methods: The init method, The str method, Operator overloading, Polymorphism. Inheritance: Importance, examples. Event handling: Key press events, Mouse events. Exceptions: Catching exceptions, Raising our own exceptions, the finally clause of the try statement.

#### **UNIT -V**

**DEFINITION AND USE OF STACKS:** Abstract data types, The Stack ADT, Implementing stacks with Python lists, pushing and popping, Using a stack to evaluate postfix, Parsing, Evaluating postfix. Queues: The Queue ADT, Linked Queue, Performance characteristics, Improved Linked Queue, Priority queue.

#### **Text Books:**

- 1) Think Python - How to Think Like a Computer Scientist, Green Tea Press, Needham, Massachusetts, Allen Downey, Version 2.0.13, June 2014.
- 2) How to Think Like a Computer Scientist: Learning with Python 3, Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, Documentation Release 3rd Edition.

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

**2 Year MCA - I SEM**

<b>L</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>3</b>	<b>3</b>

**ADVANCED DATABASE MANAGEMENT LAB**

**(14MCA21P01)**

**COURSE OBJECTIVES:**

- *Explain how a database system differs from a file system.*
- *List the main functions of a database management system*
- *Differentiate between data and information.*
- *Define the following key terms: database, Database Management System (DBMS), metadata, data inconsistency, query, single user database, multi-user database, enterprise database, centralized database, distributed database, operational database, data warehouse, data integrity, data anomaly, query language, Structured Query Language (SQL).*
- *Describe briefly the different types of databases and explain why they are important.*
- *Explain the importance of database design.*
- *Explain the terms structural dependence and data dependence.*
- *Differentiate between logical and physical data formats.*
- *List and briefly describe the main components of a database system.*
- *Define the term data redundancy and explain its effects on the quality of information produced.*
- *Differentiate among update anomaly, insertion anomaly, and deletion anomaly.*

**COURSE OUTCOMES:**

- *Explain the concepts of relational database management system (RDBMS), particularly:*
- *What an RDBMS is, and how it differs from older flat file systems.*
- *The importance of the data model, its building blocks, and how it relates to business rules.*
- *How data is organized through the use of integrity rules and primary and foreign keys.*
- *The importance of relational set operators, the data dictionary, and indexes.*
- *Explain the fundamental differences between logical and physical database design.*

- *Explain higher normal form designs, denormalized designs, existence dependence, relationship strength, weak entities, relationship participation, relationship degree, recursive relationships, and composite entities.*
- *Develop an Entity Relationship Model with the appropriate entities, attributes, relationships, connectivity, and cardinality using Crow's Foot notation to represent 1-1, 1-M, and M-N relationships.*
- *Produce a Third Normal Form database design to remove redundancies in tables.*
- *Convert the Third Normal Form database design into a set of database tables using an appropriate tool such as Oracle or MS SQL Server, and SQL Data Definition Language (DDL).*
- *Use SQL Data Manipulation Language to create and query sample data.*

#### **Recommended Systems/Software Requirements:**

- Intel based desktop PC
- Mysql /Oracle latest version Recommended

#### **List of Experiments**

1) Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

2) Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.

Example:- Select the roll number and name of the student who secured fourth rank in the class.

3) Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

4) Queries using Conversion functions (to\_char, to\_number and to\_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next\_day, add\_months, last\_day, months\_between, least, greatest, trunc, round, to\_char, to\_date)

5) i) Creation of simple PL/SQL program which includes declaration section, executable section and exception – Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)

ii) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

6) Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

7) Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.

8) Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.

9) Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.

10) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.

11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.

12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

**References:**

- 1) Database Management Systems, Peter Rob, A.Ananda Rao and Carlos Coronel, Cengage Learning.
- 2) ORACLE PL/SQL by example. Benjamin Rosenzweig, Elena Silvestrova, Pearson
- 3) Education 3rd Edition
- 4) SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande.
- 5) Introduction to SQL, Rick F.Vander Lans, Pearson Education.
- 6) Oracle PL/SQL Programming,Steven Feuerstein,SPD.
- 7) The Database Book, N.Gehani, Universities Press.
- 8) Database Systems using Oracle: A Simplified Guide to SQL and PL/SQL, Shah, PHI.

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

**2 Year MCA - I SEM**

<b>L</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>3</b>	<b>3</b>

**DESIGN AND ANALYSIS OF ALGORITHMS LAB  
(14MCA21P02)**

**Course Objectives:**

- *To analyze worst-case running time of algorithms and understand fundamental algorithmic problems.*
- *To understand how asymptotic notation is used to provide a rough classification of algorithms, how a number of algorithms for fundamental problems in computer science and engineering work and compare with one another.*
- *To introduce the methods of designing and analyzing algorithms*
- *To study about various designing paradigms of algorithms for solving real world problems.*

**Course Outcomes:**

- *To analyze the complexities of various problems in different domains.*
- *To prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains.*
- *To understand methods for analyzing the efficiency and correctness of algorithms (such as exchange arguments, recurrence, induction, and average case analysis)*
- *To design algorithms using the dynamic programming, greedy method, Backtracking, Branch and Bound strategy, and recite algorithms that employ this strategy*
- *To compare, contrast, and choose appropriate algorithmic design techniques to present an algorithm that solves a given problem.*
- *To Identify and analyze criteria and specifications appropriate to new problems.*
- *To develop the efficient algorithms for the new problem with suitable designing techniques.*
- *To know the appropriate algorithmic design technique to specific problems.*

**List of Experiments:**

1. To implement Insertion sort using array as a data structure and analyse its time complexity.
2. To implement Merge sort using array as a data structure and analyse its time complexity.
3. To implement Quick sort using array as a data structure and analyse its time complexity.
4. To implement Randomized Quick sort using array as a data structure and analyse its time complexity.
5. To implement Bubble sort using array as a data structure and analyse its time complexity.
6. To implement Bucket sort using array as a data structure and analyse its time complexity.
7. To implement Radix sort using array as a data structure and analyse its time complexity.
8. To implement Shell sort using array as a data structure and analyse its time complexity.
9. To implement Counting sort using array as a data structure and analyse its time complexity.
10. To implement Selection sort using array as a data structure and analyse its time complexity.
11. To implement Heap sort using array as a data structure and analyse its time complexity.
12. To implement Exchange sort using array as a data structure and analyse its time complexity.
13. To implement sorting algorithm of your own choice using array as a data structure and analyse its time complexity.
14. To implement Linear search and analyse its time complexity.
15. To implement Binary search and analyse its time complexity.
16. To implement Recursive Binary search and analyse its time complexity.
17. To implement Fibonacci search and analyse its time complexity.
18. To implement searching algorithm of your own choice and analyse its time complexity.
19. To implement Matrix Multiplication and analyse its time complexity.
20. To implement Breadth First Search and analyse its time complexity.
21. To implement Depth First Search without backtracking and analyse its time complexity.
22. To implement Depth First Search with backtracking and analyse its time complexity.
23. To implement Best First Search and analyse its time complexity.
24. To implement Kruskal's algorithm and analyse its time complexity.
25. To implement Prim's algorithm and analyse its time complexity.
26. To implement Tower of Hanoi problem and analyse its time complexity.
27. To implement n-Queens problem and analyse its time complexity.
28. To sort both an array and a linked list using any sorting algorithm and compare their time complexity.
29. To implement 8-puzzle problem and analyse its time complexity.
30. To implement Travelling Salesman problem and analyse its time complexity.

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE**

**(UGC - AUTONOMOUS)**

**2 Year MCA - I SEM**

<b>L</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>3</b>	<b>3</b>

**PROGRAMMING TO PYTHON LAB  
(14MCA11P03)**

***Course Objectives:***

- 1) To develop skills to design and analyze the problems.*
- 2) Students should be able to design a flowchart for any real time problem.*
- 3) To inculcate the programming skills of python language.*
- 4) To make the students to feel comfort with the object oriented programming.*

***Course Outcomes:***

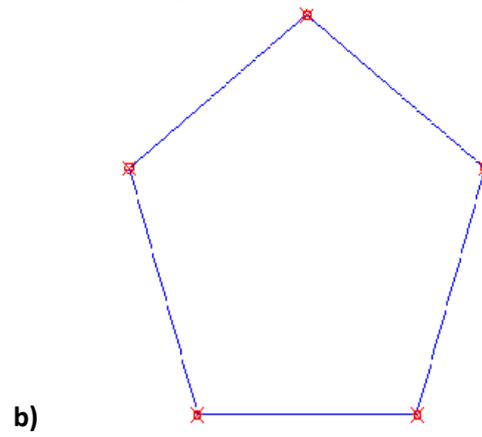
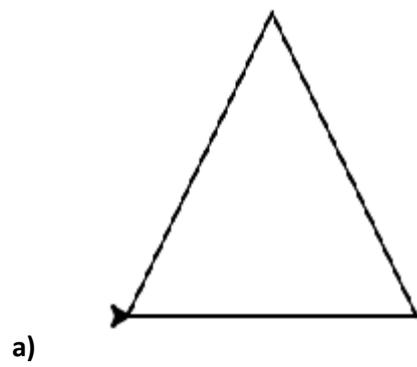
- 1. Students will able to design flowcharts for any specific problem.*
- 2. Students can learn python language constructs.*
- 3. Students will learn OOP concepts which will be helpful for them in further semesters.*
- 4. Students will have basic idea on data structures.*

**LIST OF EXPERIMENTS**

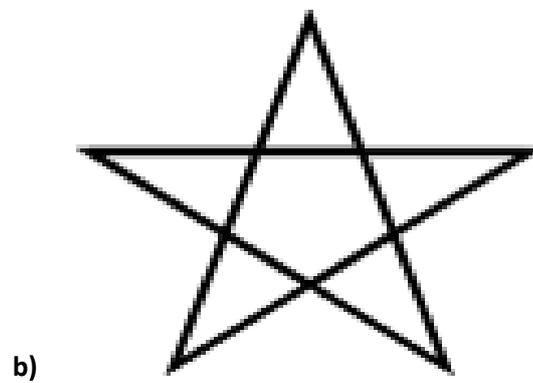
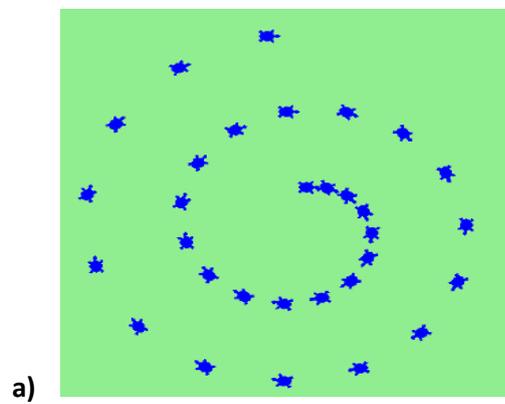
1. Develop animated models using scratch tool
2.
  - a. Develop the flowchart for finding a number is even or odd.
  - b. Develop a flowchart for displaying reversal of a number.
  - c. Develop a flowchart for finding biggest number among three numbers
3.
  - a. Develop a flowchart for swapping two values using functions.
  - b. Develop a flowchart to sort the list of numbers.
  - c. Develop a flowchart to find largest element in an array.
4.
  - a. Implement Python script to read person's age from keyboard and display whether he is eligible for voting or not.
  - b. Implement Python script to find biggest number between two numbers.
5.
  - a. Implement Python Script to generate prime numbers series up to n.
  - b. Implement Python Script to check given number is palindrome or not.
  - c. Implement Python script to print factorial of a number.
- 6.

- a. Implement Python Script to perform various operations on string using string libraries.
  - b. Implement Python Script to check given string is palindrome or not.
  
- 7.**
  - a. Define a function `max_of_three()` that takes three numbers as arguments and returns the largest of them.
  - b. Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.
  
- 8.**
  - a. Define a function which generates Fibonacci series up to n numbers.
  - b. Define a function that checks whether the given number is Armstrong.
  
- 9.**
  - a. Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.
    - a. Suppose the following input is supplied to the program:34,67,55,33,12,98. Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34', '67', '55', '33', '12', '98').
  - b. With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program to print the first half values in one line and the last half values in one line.
  
- 10.**
  - a. Write a python script to perform basic dictionary operations like insert, delete and display.
  - b. Write a python script to find frequency of words in a file using dictionaries.
  
- 11.**
  - a. Write Python script to display file contents.
  - b. Write Python script to copy file contents from one file to another.
  
- 12.**
  - a. Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.
  - b. Define a class named Circle which can constructed by radius. The derived classes Area, Circumference uses methods called `calArea()`, `calCirc()` respectively to calculate area, circumference of circle.
  
- 13.**
  - a. Implement Python script to develop stack ADT and its operations.
  - b. Implement Python script to evaluate postfix expression.
  
- 14.**
  - a. Implement Python script to develop queue ADT and its operations.
  - b. Implement Python script to perform tree traversals.

15. Write a python script to display following shapes using turtle.



16. Write a python script to display following shapes using turtle.



**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

**2 Year MCA - II SEM**

<b>L</b>	<b>T</b>	<b>C</b>
<b>4</b>	<b>0</b>	<b>4</b>

**SOFTWARE ENGINEERING**

**(14MCA22T06)**

**Course Objectives:**

- *The purpose of project management is to foresee or predict as many dangers and problems as possible; and to plan, organise and control activities so that the project is completed as successfully as possible in spite of all the risks.*
- *The ever-present element of risk and uncertainty means that events and tasks leading to completion can never be foretold with absolute accuracy. For some complex or advanced projects, even the possibility of successful completion might be of serious doubt.*
- *Project management can involve the following activities: planning - deciding what is to be done; organising - making arrangements; staffing - selecting the right people for the job; directing - giving instructions; monitoring - checking on progress; controlling - taking action to remedy hold ups; innovation - coming up with new solutions; representing - liaising with users.*
- *Project Management is generally seen as a key component of successful software projects. Together with software techniques it can produce software of high quality. This course aims to cover the basics*
- *Deliver successful software projects that support organization's strategic goals*
- *Match organizational needs to the most effective software development model*
- *Plan and manage projects at each stage of the software development life cycle (SDLC)*
- *Create project plans that address real-world management challenges*
- *Develop the skills for tracking and controlling software deliverables*

**Course Outcomes:**

- *Should be able to know basic terms associated with system and system analysis, such terms are SDLC System Development Life Cycle and System Thinking*
- *To understand the alternatives of project organizations and advantages of each in order to choose the appropriate one for the project to achieve its goals.*
- *Understand the process of managing information system project.*
- *Should be able to learn the skilled needed by a project Manger to be affective, professional and a successful individual*
- *To learn how to adapt to new circumstance and keep project on target as close to plan as feasible*
- *To understand new ways to improve costing and estimation of projects.*

**UNIT- I**

**Software, Software Engineering, and Process:** The nature of Software, The unique nature of WebApps, Software engineering- A layered technology, The essence and principles of software engineering practice, Generic process model (framework), Process patterns, Process assessment and improvement, CMMI, Software myths.

**Process Models:** Prescriptive process models: The waterfall model, Incremental process models, Evolutionary process models. The Unified process, Aspect oriented software development, Agile development: Agile process, Extreme programming.

## UNIT- II

**Software Requirements:** Introduction to functional and non-functional requirements, Requirements engineering activities, Eliciting requirements, Requirements modeling, Requirements validation, Software requirements specification (SRS), Requirements management.

**Requirements modeling:** Structured view: Data modeling (ERD), Flow-Oriented modeling (DFD), Behavioral modeling, Object models, and structured methods. **Software Project Estimation:** Empirical estimation models.

**Design Concepts:** Software design quality guidelines and attributes, Design concepts, Design model.

**Software Architecture:** Architecture and its importance, Architectural Styles, Data design, Architectural design

**Design:** Structured view (Traditional view): Architectural mapping using data flow (Call and return architecture), Interface design, Function based component design.

**Object oriented view:** OO Architecture, Class hierarchies, Message design, Class based component design.

## UNIT- III

**Performing User Interface Design:** Golden rules, User interface analysis and design, interface analysis, interface design steps.

**Pattern Based Design:** Design patterns, Pattern based software design, Architectural patterns, Component level design patterns, User interface design patterns.

## UNIT- IV

**Testing:** Software testing strategies: A strategic approach to software testing, Test strategies (Unit testing and integration testing) for conventional and object oriented software, Validation testing, System testing, The art of debugging.

**Testing Conventional Applications:** Software testing fundamentals, White-Box testing: Basis path testing, condition (predicate) testing, data flow testing, loop testing, Black box testing: Equivalence partitioning, Boundary value analysis, Graph based testing methods.

**Testing Object Oriented Applications:** OO testing methods, Testing methods applicable at class level, Interclass test case design.

## UNIT-V

### **Umbrella Activities :**

Risk management, Software quality assurance, Software configuration management, Measurement and metrics: Size oriented metrics, Function oriented metrics, Metrics for software quality, Product metrics: Metrics for the requirements model, Metrics for the design model, Metrics for source code, Metrics for testing, Metrics for maintenance.

**Software Reengineering:** A software reengineering process model, Software reengineering activities.

### **Text Books:**

- 1) Software Engineering, A practitioner" s Approach- Roger S. Pressman, 7th edition. McGrawHill International Edition.

- 2) Software Engineering- Sommerville , 8th edition, Pearson education.
- 3) . Software Engineering- K.K. Agarwal & Yogesh Singh,New Age International Publishers
- 4) Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiely.

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE**  
**(UGC - AUTONOMOUS)**

2 Year MCA - II SEM

**WEB PROGRAMMING THROUGH PHP**  
**(14MCA22T07)**

L	T	C
4	0	4

**Course Objectives:**

*The main objectives of the course are summarized as below:*

- *Students will be introduced to a number of topics in PHP programming and its environment.*
- *Students will get exposed to a data types, Conditional statements, loops, arrays available in PHP.*
- *Course will give exposure to important technologies and concepts. (E.g. MySQL Database, PHP functions, File I/O, regular expressions etc.).*
- *Students will be introduced to HTML Forms and how PHP can work with HTML Forms together.*
- *Students of the Course will be introduced to XML and PHP XML Parsers.*

**Course Outcomes:**

*The aim of this course is to equip you with the knowledge you will need to build basic Web sites.*

*By the end of this course student should be able to:*

- *Do basic programming and advanced server-side scripting using PHP.*
- *Recognize the difference between HTML, XML, MySQL, and PHP.*
- *Differentiate between PHP Web & HTML Controls*
- *Understand different Web controls*
- *Understand connecting Web pages with MySQL DB.*
- *Students will be able to develop a complete end-to-end web applications using PHP.*

**UNIT- I:**

**PHP Introduction:**PHP,Benefits of using PHP MySQL,Setup of PHP Environment,Testing the Page,Troubleshooting Installation Errors

**PHP Programming Concepts:**Write your First PHP Program,Embed PHP in HTML / HTML in PHP,PHP Data Types,Variables in PHP,SuperGlobal Variables,Operators in PHP,Conditional Statements,Loops (For, While, Do While, Foreach)

**UNIT- II**

**MySQL Database Training:** What is Database? MySQL Database MySQL Connect MySQL Create DB MySQL Create Table MySQL Insert Data MySQL Get Last ID MySQL Insert Multiple MySQL Prepared MySQL Select Data MySQL Delete Data MySQL Update Data MySQL Limit Data.

**UNIT- III**

**Array:** Why use Arrays, Types of Arrays, Creating Arrays, Accessing Arrays, Array Functions, Using Array Functions, Enumerated Arrays : Initializing Arrays, Appending to an Array, Reading from Arrays, Looping through Arrays, Associative Arrays: Initializing Associative Arrays, Reading from Associative Arrays, Looping through Associative Arrays, Superglobal Arrays, Two-dimensional Arrays: Reading from Two-dimensional Arrays, Looping through Two-dimensional Arrays

**PHP Functions:** Using Functions in PHP, Userdefined Functions, Predefined Functions, Common Functions, String Functions, File Functions, Date Functions, Hash Functions, Mail Function

#### **UNIT -IV**

“File I/O and the File System,” delves into one of PHP’s noted features, manipulation of data files. Information is also provided regarding how PHP can interact with the many facets of server directories.

“Strings and Regular Expressions,” discusses string manipulation through the use of predefined functionality and regular expressions. Both POSIX and Perl-type regular expressions are introduced.

#### **UNIT –V**

PHP and HTML Forms : HTML Forms , How HTML Forms Work, A Sample HTML Form, Form Variables, String Manipulation , Formatting Strings : Concatenation, String Manipulation Functions, Examples of String Functions, “PHP and Dynamic Site Development,” primes the reader on the very basic concepts of using PHP to create dynamic Web sites. Dynamic content, link creation, and basic page templating strategies are covered in this chapter. , “Forms,” describes how PHP can work with HTML forms to gather, display, and manipulate user input.

**PHP and XML,:** PHP XML Parseers, PHP SimpleXML Parser, PHP SimpleXML-Get, PHP XML Expat, PHP XML DOM

#### **Text Books:**

- 1) Php: The Complete Reference (English) 1st Edition by Steven Holzner.
- 2) Head First PHP & MySQL (English) 1st Edition by Michael Morrison and Lynn Beighley.
- 3) Beginning PHP6, Apache, Mysql Web Development (English) by Timothy Boronczyk Jason Gerner , Jeremy Stolz , Elizabeth Naramore and Yann Le Scouarnec.
- 4) Beginning PHP and MySQL: From Novice to Professional (English) 4th Edition by W. Jason Gilmore
- 5) PHP and MySQL Web Development (English) 4th Edition by Luke Welling and Laura Thomson.
- 6) Murach's PHP and MySQL: Training & Reference (English) by Joel Murach and Ray Harris

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

2 Year MCA - II SEM

L	T	C
4	0	4

**OPERATING SYSTEMS**

**(14MCA22T08)**

**Course Objective:**

*To study and apply concepts relating to operating systems:*

- *General understanding of structure of modern computers.*
- *To understand the services provided by and the design of an operating system.*
- *To understand the structure and organization of the file system.*
- *To understand what a process is and how processes are synchronized and scheduled.*
- *To understand different approaches to memory management.*
- *Students should be able to use system calls for managing processes, memory and the file system.*
- *Students should understand the data structures and algorithms used to implement an OS.*
- *Illustration of key OS aspects by example.*

**Course Out comes:**

*By the end of the course you should be able to*

- *Describe the general architecture of computers.*
- *Describe, contrast and compare differing structures for operating systems.*
- *Understand and analyse theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files.*
- *A high-level understanding of the structure of operating systems, applications, and the relationship between them.*
- *Understand FAT filing systems and NTFS filing systems.*
- *Evaluate the security issues of FAT Tables and the more secure NTFS.*
- *Prevent and repair infections of malware on a computer*
- *Design, implement, and configure physical security of computers*

**UNIT- I:**

**Operating System Introduction:** Operating Systems objectives and functions, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special -Purpose Systems, Operating System services, User OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure, Virtual Machines.

**UNIT- II:**

**PROCESS:** Overview of Process, Process Scheduling, scheduling Algorithm, Multiple-Processor Scheduling, The Critical Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization.

**Deadlocks** - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

**UNIT -III**

**Memory Management:** Logical & Physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging. Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames, Thrashing, Case Studies: Linux, Windows.

#### **UNIT- IV**

**Storage Management:** File System – File Concept, Access methods, Directory Structure, File System Mounting, File Sharing, Protection. Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management, Free-space Management.

#### **UNIT- V**

**Protection** – System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

**Security-** The Security problem, Program threats, System and network threats, Cryptography as a security tool, User authentication, Implementing security defenses, Firewalling to protect systems and networks, Computer –security classifications, Case Studies: Linux, Windows.

#### **Text Books:**

1. **Operating System Principles** , Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley Student Edition
2. **Modern Operating Systems**, Andrew S Tanenbaum, 3rd Edition, Pearson/PHI

#### **Reference Books:**

1. Operating Systems – Internals and Design Principles, W. Stallings, 6th Edition, Pearson Education.
2. Operating Systems A concept-based Approach, 2nd Edition, D.M.Dhamdhare, TMH.
3. Principles of Operating Systems , B.L.Stuart, Cengage learning, India Edition.
4. Operating Systems, A.S.Godbole,2nd Edition, TMH
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6. Operating Systems, R.Elmasri, A,G.Carrick and D.Levine, Mc Graw Hill.

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

2 Year MCA - II SEM

L    T    C  
4    0    4

**NETWORK SECURITY ESSENTIALS APPLICATIONS AND STANDARDS  
(14MCA22T09)**

**Course Objectives:**

- *To Study Security: Cipherring of plain text , Attacks, Services & Mechanisms*
- *Study of various Cryptography security algorithms and encryption methods for security and protection*
- *Study key management issues and implementation of hash function to messages*
- *To Understand Digital signature and its importance in transaction processing*
- *Study on Web security, firewalls and supporting protocols*

**Course Out comes:**

- *Understand, analyze and work on activities of fraud prevention, monitoring, investigation, reporting.*
- *Differentiate among the models, architectures, challenges and global legal constraints of secure electronic commerce technologies used to ensure transmission, processing and storage of sensitive information.*
- *Have knowledge of Digital signatures, hash functions and key management.*
- *Can understand the impact of security mechanisms and protocols on online transaction*
- *Acquire knowledge on firewalls and web security concepts*

**UNIT-I**

**INTRODUCTION:** Terminology—notation—primer on networking—types of attacks- Layer and cryptography- Authorization-Key Escrow- Viruses, worms and Trojan Horses- Multi Level mode of security- legal issues.

**UNIT-II**

**CRYPTOGRAPHY:** Introduction-Secret Key cryptography-Public Key Cryptography-Hash algorithm-DES-IDEA-AES-Modes of Operations-Hashes and Message Digests-MD2- MD4-MD5 and SHA-1-RSA-Diffie-Hellman-Digital Signature Standard (DSS)-Elliptic Curve Cryptography.

**UNIT-III**

**AUTHENTICATION:** Password based authentication-address based authentication-Cryptographic authentication Protocols-Passwords as cryptographic keys-trusted Intermediaries-certificate revocation-Multiple trusted Intermediaries-Session Key Establishment-Delegation.

**UNIT-IV**

**STANDARDS:** Kerberos V4-Kerberos V5-Public Key Infrastructure-Real Time communication Security-IPsec: AH and ESP-IPsec: IKE-SSL/TLS

**UNIT-V**

## **ELECTRONICMAIL:E-MailSecurity-PEM&S/MIMEandPGP**

### **Text Books:**

- 1) Network Security Private Communication in a public world, CharlieKaufman, RadiaPerlman&MikeSpeciner,PearsonEducation/PrenticeHallofIndiaPrivate Ltd.,NewDelhi.(Chapters:1to6,9,13to22)

### **Reference Books:**

- 1)NetworkSecurityEssentialsApplicationsandStandards,WilliamStallings,PearsonEducation,NewDelhi
- 2) CryptographyandNetworksecurity,AtulKahate,TataMcGraw-HillPub companyLtd.,NewDelhi

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE  
(UGC - AUTONOMOUS)**

2 Year MCA - II SEM

L	T	C
4	0	4

**COMPUTER ORGANIZATION**

**(14MCA22T10)**

**Course Objectives**

- *The main objective of computer organization course is to introduce the main concepts and components of computer organization and architecture.*
- *Understand the architecture of a modern computer with its various processing units.*
- *Understand the performance measurement of the computer system*
- *Understand the Cache memory and its importance.*
- *Students can calculate the effective address of an operand by addressing modes*

**Course Outcomes:**

- *Successful completion of this course should lead to the following learning outcomes:*
- *Understand the combinational and sequential logical circuits.*
- *Understand the basics of assembly language.*
- *Understand the main concepts of computer architecture.*
- *Design and analyze the main functional units of a computer.*
- *Implement assembly programs that accomplish basic computational and I/O operations.*
- *Discuss and work in a group in order to design the main functional units of a computer.*
- *Discuss and work in a group in order to design and implement assigned programming tasks in Assembly.*
- *Demonstrate designed functional units as well as implemented assembly programs.*

**UNIT- I**

**INTRODUCTION:** Overview of basic digital building blocks; basic structure of a digital computer. Number system and representation of information, arithmetic and logical operation, hardware implementation, Real numbers - fixed and floating point, IEEE754 representation. logic gates, Sequential and combinational circuits, flip flop, adders, multiplexers, encoders, decoders

**UNIT -II**

**BASIC BUILDING BLOCKS FOR THE ALU:** Adder, Subtractor, Shifter, Multiplication and division circuits. CPU : Data path - ALU, Registers, CPU buses; Control path - microprogramming (only the idea), hardwired logic, RISC, CISC;

**UNIT- III**

Instruction format, Bus, Common Bus, hand shaking, External interface. Various addressing modes. Concept of sub-routine and sub-routine call. Use of stack for handling sub-routine call and return, instruction interpretation and execution. Bus Architecture

**UNIT -IV**

**MEMORY :** Memory organization; concepts of semi-conductor memory, CPU memory interaction, organization of memory modules, cache memory and related mapping and replacement policies, virtual memory

## **UNIT -V**

**I/O Sub block: I/O techniques** - interrupts, polling, DMA; Synchronous vs. Asynchronous I/O; Controllers. – Interface circuits – Standard I/O Interfaces (PCI, SCSI, USB), I/O devices and processors. Introduction to VHDL concepts: examples to be taken up from the rest of the course for implementation.

### **Text Books:**

1. Computer Organization, Hamachar, Vranesic & Zaky.

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE**  
**(UGC - AUTONOMOUS)**

**2 Year MCA - II SEM**

<b>L</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>3</b>	<b>3</b>

**WEB PROGRAMMING THROUGH PHP LAB**  
**(14IMCA22P05)**

**Course Objectives:**

The main objectives of the course are summarized as below:

- Students will be introduced to a MySQL Database, CGI and Python programming.
- Students will put their programming skills to use in constructing a complete end to-end application combining PHP, Python, and MySQL DB etc.
- Students will be introduced to advanced topics such as XML, XML Parsers etc.

**Course Outcomes:**

By the end of this course student should be able to:

- Understand different control structures available in PHP, Python.
- Understand connecting Web pages with MySQL DB.
- Develop a complete end-to-end web applications using PHP combining Python, MySQL DB with PHP.

**List of Experiments**

1. Write a Program to demonstrate simple control structures.
2. Write a program to demonstrate different types of lists
3. Write a Program to demonstrate exception handling
4. Explain the procedure to install SQLite
5. How SQLite is installed on Mac and create a sample database
6. Explain inserting, deleting and selecting data.
7. Explain DB programming with an example
8. How data is transferred within files using FTP.

9. Explain Python Email program with an example.
10. Explain web page structure and web client programming with python with examples
11. Explain using urlilib using with an example
12. Write a program to demonstrate parsing HTML
13. Write a program to demonstrate screen scraper and web crawler
14. Write a program to demonstrate CGI programming
15. Explain form processing with an example
16. Explain input database data and update database data and various database applications in python with examples.
17. Write a program to demonstrate XML programming with python
18. Write a program to demonstrate Django web application framework (setting Django and Django API).

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE**  
**(UGC - AUTONOMOUS)**

**2 Year MCA - II SEM**

<b>L</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>3</b>	<b>3</b>

**OPERATING SYSTEM LAB**

**(14IMCA22P05)**

**Course Objective**

- Analyzing the working of an operating system and its components
- Defining and Analyzing the synchronization process
- To understand the design aspects of operating system.
- To Study the management of different type of memories in the computer system
- Comparing and analyzing different file systems being used in different operating systems

**Course Outcome**

- To use Unix utilities and perform basic shell control of the utilities
- To use the Unix file system and file access control.
- To implement process scheduling & synchronization algorithms
- To implement the application by using inter process communication
- To implement memory management scheme & file allocation like best fit, worse fit etc.

**List of Experiments**

(Implement the following on LINUX or other Unix like platform. Use C for high level language implementation)

1. Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close, stat, opendir, readdir
2. Write programs using the I/O system calls of UNIX operating system (open, read, write, etc)
3. Write C programs to simulate UNIX commands like ls, grep, etc.
4. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)
5. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)
6. Developing Application using Inter Process communication (using shared memory, pipes or message queues)

7. Implement the Producer – Consumer problem using semaphores (using UNIX system calls).
8. Implement some memory management schemes – I
9. Implement some memory management schemes – II
10. Implement any file allocation technique (Linked, Indexed or Contiguous)

Example for exercises 8 & 9 :

Free space is maintained as a linked list of nodes with each node having the starting byte address and the ending byte address of a free block. Each memory request consists of the process-id and the amount of storage space required in bytes. Allocated memory space is again maintained as a linked list of nodes with each node having the process-id, starting byte address and the ending byte address of the allocated space. When a process finishes (taken as input) the appropriate node from the allocated list should be deleted and this free disk space should be added to the free space list. [Care should be taken to merge contiguous free blocks into one single block. This results in deleting more than one node from the free space list and changing the start and end address in the appropriate node]. For allocation use first fit, worst fit and best fit.

2 Year MCA - II SEM

L	P	C
0	3	3

**NETWORK SECURITY LAB**

**(14IMCA22P06)**

**Course Objectives:**

*This course provides an in-depth look into the theoretical and technical aspects of risks related to computer networked systems. The course focuses on the topics of network security, encryption and decryption techniques, network environments, security policy and planning, secure networks, access control, firewalls, socket programming, application computer network security in small business organizations for data protection. All the above topics will include theoretical and technical background, as well as hands on practical experience.*

**Course Outcome:**

*The purpose of this course is to provide an introductory level concepts and hands-on practical experience to the students. On successful completion of this course, student will be able to say*

- *Concepts of computer networks and issues concerned with computer network security.*
- *Strong practical knowledge in encrypting the plain text, and converting it into cipher text using classical encryption techniques.*
- *Exploring the possible attacks in networks and prevention using SSH, Comparison of various network architectures.*
- *Theoretical & Technical concepts of WLAN. Constraints on WLAN and traffic usage VPN Connection.*
- *Construction of VPN over WLN, understanding the security and risks related to small business organization setup.*
- *Understanding the need of policies for security management in client-server architecture.*
- *Understanding the technical concepts on binary / byte encryption.*
- *Implementation of Sub netting and OSI model and VLAN logical groups with virtual LAN's.*

**List of Experiments**

**1. Encryption Using Classical Techniques**

In this project you will develop a program to encrypt plaintext text given a keyword. The plaintext will be encrypted by Playfair cipher and the cipher text is displayed for a user.

Playfair Cipher (description taken from William Stallings "Cryptography and Network Security, Principles and Practice) is the best-known multiple letter encryption cipher, which treats diagrams in the plaintext as single units and translates these units into cipher text diagrams. (This cipher was actually invented by British scientist Sir Charles Wheatstone in 1854, but it bears the name of his friend Baron Playfair of St. Andrews, who championed the cipher at the British foreign office.)

The Playfair algorithm is based on the use of a 5 x 5 matrix of letters constructed using a keyword. Here is an example:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

In this case, the keyword is monarchy. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter. Plaintext is encrypted two letters at a time according to the following rules:

1. Repeating plaintext letters that would fall in the same pair are separated with a filler letter, such as x, so that balloon would be enciphered as ba lx lo on.
2. Plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example ar, is encrypted as RM.
3. Plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the row circularly following the last. For example, mu is encrypted as CM.
4. Otherwise, each plaintext letter is replaced by the letter that lies in its own row and column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM).

## 2.Eavesdropping Attacks and its prevention using SSH

The goal of this experiment is to teach student

1. Design and implementation of a simple client/server model and running application using sockets and TCP/IP.
2. To make students aware of the insecurity of default passwords, printed passwords and password transmitted in plain text.
3. To teach student how to use SSH for secure file transfer or for accessing local computer using port forwarding technique.
4. Comparison between Telnet and SSH Configuration

To start with, student first connects two stations on a switch/hub and a server on the same. The stations has SSH client installed on it and other port scanning software such as Ethereal, free port scanner etc. The server is configured as simple file server which has one network drive to share important documents and common software executable which can be accessed on clients. The server is configured for DHCP and for Routing and remote access for internet. The configuration has two separate network adapters, one which has DHCP running on it and second has access to Internet gateway. The routing and remote access configuration is customized to route packets between these two interfaces. The NATed packet flows over this routing. The server is configured for secure access of entire system and is done by security configuration wizard. The security configuration wizard asks for port number to be left open on startup.

Here, port 22, port 23, port 80 and other ports are kept open all the time. The server has third party SSH demon running all the times and has interface with command prompt for configuration and control.

### Problem

Computer communication is a transfer of information between two computers. We use log in name and password to secure access from one computer to another. If this log in name and password is listened by some unwanted user on the network, it can access that same computer with same privileges as first one.

We do have password protection security but in spite of that hackers are successful in stealing the secure information. This experiment will explain how can we secure this communication more with simple SSH client and make it robust from any unwanted attack.

## 1) Comparison of Telnet and SSH

### Telnet Connection

- 1) Open Telnet client on one of the terminal.
- 2) Start ethereal packet sniffer on server as well as client.
- 3) Start port scanner on server and monitor Telnet ( port 23)
- 4) Connect client to server using Telnet, enter username and password, and stop ethereal packet sniffer on both server and client. Observation

- 1) On both ethereal we can see number of frames transmitted and received as well as protocol used, port number etc.
- 2) We can also see the actual data sent on real time basis and corresponding reception on server/client.

### Conclusion

- 1) Observe port number on which client is working and port number on server is working.
- 2) Observe number of packets require to complete login process.
- 3) Observe any security certificate or encryption used
- 4) Identify username and password used to log in on server

### SSH connection

- 1) Open SSH client on one of the terminal.
- 2) Start ethereal packet sniffer on server as well as client.
- 3) Start port scanner on server and monitor SSH ( port 22)
- 4) Connect client to server using SSH, enter username and password, and stop ethereal packet sniffer on both server and client.

### Observation

- 1) On both ethereal we can see number of frames transmitted and received as well as protocol used, port number etc.
- 2) We can also see the data sent from server to client and vice-versa.
- 3) The entire data frame is encrypted and ASCII characters are displayed on the screen.
- 4) SSH protocol uses Key Exchange algorithm for Authentication and AES/DES encryption for ciphering the text i.e. data.
- 5) After network layer, no actual data is visible.

### Conclusion

- 1) Observe port number on which client is working and port number on server is working.
- 2) Observe number of packets require to complete login process.
- 3) Observe and note down specific key exchange algorithm used by SSH.
- 4) Observe the Hash function used by AES/DES Cipher algorithm, it is either 128 bit or 256 bits
- 5) Try to recover actual data by attacking the algorithm

The above method can be compared again by connecting hub instead of switch.

### SSH port forwarding

The SSH protocol has the ability to forward arbitrary network connections over your encrypted SSH connection, to avoid the network traffic being sent in clear. For example, you could use this to connect from your home computer to a POP-3 server on a remote machine without your POP-3 password being visible to network sniffers.

In order to use port forwarding to connect from your local machine to a port on a remote server, you need to following settings.

- Choose a port number on your local machine where SSH client (PuTTY) should listen for incoming connections. There are likely to be plenty of unused port numbers above 3000
- Now, before you start your SSH connection, go to the Tunnels panel. Make sure the 'Local' radio button is set. Enter the local port number into the 'Source port' box. Enter the destination host name and port number into the 'Destination' box, separated by a colon (for example, www.csulb.edu:80 to connect to a web server).
- Now click the 'Add' button. The details of your port forwarding should appear in the list box.

Fig. 1 SSH Port Forwarding diagram Step 1:- Enter the IP address of server in Host name Box. Port 22 is for SSH

Step 2:- Use any source port above number 3000 (e.g. 5050) and give the destination address and port number of remote machine (e.g. www.csulb.edu:80)

Step 3:- Click open to connect to server and type in correct log in name and password to get the access. Observation

1) Go to web browser on machine and type http://localhost:5050 in address bar. You will see www.csulb.edu website opened on port 5050.

2) Open ethereal window to see what packet we have received on sniffer

3) You can see http traffic on clients screen but data is not visible neither port 5050 is visible anywhere on ethereal. This means sniffer don't even know that the port 5050 is open and we are browsing internet on it.

4) On server side you can see the port (80) is open which is for http traffic.

### 3. Isolating WLAN Traffic using Separate Firewall for VPN Connection

Enterprise and businesses of all sizes are concerned with corporate data being exposed due to lapses in wireless device security. With expansions to a mobile user base, companies must recognize wireless security as a valid concern. Early WLANs frequently re-used remote access VPN clients to overcome the limitations of WEP and related security concerns. But, given improvements in Wi-Fi security, does VPN still a secure communication in enterprise wireless? What are the practical benefits and limitations of using VPN over wireless? Is it all the VPN client software safe to connect? We know that big enterprise offer laptop to its employee to provide work mobility. These laptops roam around different network accessing different information on the internet. This wide access to internet puts virus, Trojans and spyware to settle on the enterprise laptops and later on when it connects to enterprise network; it can access valuable data on the server. Further, if the same laptop makes a VPN connection with remote site, it is likely to access the remote information once it is fetched on laptop. This problem is related to site-to-site VPN connection and could be fatal if proper security procedure is not followed.

### 4.Virtual Private Network Over WAN

The goal of this experiment is

1)To Create a Virtual Private Network (VPN) over WAN

2) To evaluate application response time in the presence and absence of a firewall.

#### **Managing Security in Small Business Network**

Server/client networks are important to establish when the number of computers in your network increases and therefore it becomes impossible for peer to peer networks to handle all the computing .In such model, security is maintained easier, since the server is responsible for managing all the other users .In server based environment, users and computers are created and granted access according to the administrator's needs .Different rights are assigned to users and computers, giving the ability for the users /computers to print, use internet etc... In this lab , a small business network is established in which an active directory is installed and users and computers are added to form a base for the network security .In order to acquire a better understanding of how an active directory works

### 5. Security Group Policies Management

Group Policy management is a Windows Server 2003 features in which it allows administrators to define policies for both servers and user machines .Group policies ensure that all computers in the network are secure and ease the process of deployment by keeping the network safe . They allow the definition of user and computer related policies, network settings and security settings. In this lab, we will go over implementing group policy object, configuring group policies and determining applied group policies objects. In addition, we are going to apply different security policies to groups.

### 6. Lab on Frequency Analysis

The cryptanalyst can benefit from some inherent characteristics of the plaintext language to launch a statistical attack. For example, we know that the letter E is the most frequently used letter in English text. The cryptanalyst finds the mostly-used character in the ciphertext and assumes that the corresponding plaintext character is E. After finding a few pairs, the analyst can find the key and use it to decrypt the message. To prevent this type of attack, the cipher should hide the characteristics of the language. Table 1 contains frequency of characters in English.

Table 1 Frequency of characters in English

<i>Letter</i>	<i>Frequency</i>	<i>Letter</i>	<i>Frequency</i>	<i>Letter</i>	<i>Frequency</i>	<i>Letter</i>	<i>Frequency</i>
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

### 7. Encryption using binary/byte addition

Under this encryption algorithm, the key entered is added character by character (byte by byte) to the data to be encrypted. Here addition modulo 256 is used, i.e. so that any carry-overs are ignored. The key is applied cyclically (as under the Vigenère encryption algorithm and also with the Exclusive-OR), i.e. once all the characters (bytes) of the key have been used, the algorithm reverts to the first character until the text has been completely encrypted.

To decrypt the text, the characters of the key have to be subtracted from the encrypted text modulo 256.

If one knows the characters which occur most frequently in the plaintext, it is then possible to work out the key with the aid of a computer (and hence also the plaintext) (see Automatic analysis, Byte Addition).

The key used for Binary Addition is entered in the Key entry dialog.

This encryption algorithm can be easily broken with a Ciphertext-Only attack (see Automatic analysis, Byte Addition).

### 8. Implementing ICMP Ping

### 9. Implementation of Sub netting and OSI model.

### 10. Implementation of VLAN logical groups with virtual LAN's.