LINGAYA'S VIDYAPEETH
SCHEME OF STUDIES
SESSION: 2022-23

School: School Of Computer Science & Information Technology	**Batch: 2022-2024**
Department: Department of Computer Applications	**Year: 1st**
Course: Master of Computer Applications (MCA)	**Semester: I**

SN	**Category**	**Course Code**	**Course Name**	**Periods**			**Credits**	**Evaluation Scheme**					**Subject Total Marks**
								Theory			**Practical**		
				L	**T**	**P**		**ABQ**	**MSE**	**ESE**	**IP**	**EXP**	
1	PCC	MCA-109	Programming and Problem-Solving using C	3	0	0	3	15	25	60	-	-	100
2	PCC	MCA-105	Computer Networks	3	1	0	4	15	25	60	-	-	100
3	PCC	MCA-111	Digital Logic & Computer Architecture	3	1	0	4	15	25	60	-	-	100
4	PCC	MCA-113	Data Structure and Algorithm	3	0	0	3	15	25	60	-	-	100
5	PCC	MCA-115	Database Management Systems	3	0	0	3	15	25	60	-	-	100
6	PCC	MCA-163	Data Structure and Algorithm Lab	0	0	2	1	-	-	-	60	40	100
7	PCC	MCA-159	Programming and Problem-Solving using C Lab	0	0	2	1	-	-	-	60	40	100
8	PCC	MCA-165	DBMS Lab	0	0	2	1	-	-	-	60	40	100
9	PCC	MCA-175	Project Centric approach	0	0	4	2	-	-	-	-	100	100
			Total---->	**15**	**2**	**10**	**22**						

LINGAYA’S VIDYAPEETH

SCHEME OF STUDIES

SESSION: 2022-23

School: School Of Computer Science & Information Technology	**Batch: 2022-2024**
Department: Department of Computer Applications	**Year: 1st**
Course: Master of Computer Applications (MCA)	**Semester: II**

SN	**Category**	**Course Code**	**Course Name**	**Periods**			**Credits**	**Evaluation Scheme**					**Subject Total Marks**
								Theory			**Practical**		
				L	**T**	**P**		**ABQ**	**MSE**	**ESE**	**IP**	**EXP**	
1	PCC	MCA-112	Software Engineering & Testing	3	1	0	4	15	25	60	-	-	100
2	PCC	MCA-114	Object Oriented Programming using C++	3	0	0	3	15	25	60	-	-	100
3	PCC	MCA-116	Operating System & Shell Programming	3	1	0	4	15	25	60	-	-	100
4	PCC	MCA-118	Artificial Intelligence	3	0	0	3	15	25	60	-	-	100
5	PCC	MCA-120	Statistical & Numerical Computing	3	0	0	3	15	25	60	-	-	100
6	PCC		Elective-I	3	0	0	3	15	25	60	-	-	100
7	PCC	MCA-164	Object Oriented Programming Lab	0	0	2	1	-	-	-	60	40	100
8	PCC	MCA-168	Artificial Intelligence Lab	0	0	2	1				60	40	100
9	PCC	MCA-170	Statistical & Numerical Computing Lab	0	0	2	1				60	40	100
			Total---->	**18**	**2**	**6**	**23**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
AECC:	Ability Enhancement Compulsory course	ESE:	End Semester Examination
GE:	General English	IP:	Internal Practical
DSE:	Discipline Specific Elective Course	EXP:	External Practical
L:	Lecture	PROJ:	Project
T:	Tutorial		

LINGAYA’S VIDYAPEETH

SCHEME OF STUDIES

SESSION: 2023-24

School: School Of Computer Science & Information Technology	**Batch: 2022-2024**
Department: Department of Computer Applications	**Year: 2nd**
Course: Master of Computer Applications (MCA)	**Semester: III**

SN	**Category**	**Course Code**	**Course Name**	**Periods**			**Credits**	**Evaluation Scheme**					**Subject Total Marks**
								Theory			**Practical**		
				L	**T**	**P**		**ABQ**	**MSE**	**ESE**	**IP**	**EXP**	
1	PCC	MCA-209	Operational Research & Optimization	3	0	0	3	15	25	60	-	-	100
2	PCC	MCA-211	Digital Image processing	3	0	0	3	15	25	60	-	-	100
3	PCC	MCA-205	Python Programming	3	0	0	3	15	25	60	-	-	100
4	PCC	MCA-213	Soft Computing	3	0	0	3	15	25	60	-	-	100
5	PCC	MCA-215	Introduction to Machine Learning for Data Science	3	0	0	3	15	25	60	-	-	100
6	PCC		**Elective-II**	3	0	0	3	15	25	60	-	-	100
7	PCC	MCA-263	Soft Computing Lab	0	0	2	1				60	40	100
8	PCC	MCA-255	Python Programming Lab	0	0	2	1				60	40	100
9	PCC	MCA-265	Introduction to Machine Learning for Data Science Lab	0	0	2	1				60	40	100
			Total---->	**18**	**0**	**6**	**21**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
AECC:	Ability Enhancement Compulsory course	ESE:	End Semester Examination
GE:	General English	IP:	Internal Practical
DSE:	Discipline Specific Elective Course	EXP:	External Practical
L:	Lecture	PROJ:	Project

LINGAYA'S VIDYAPEETH

SCHEME OF STUDIES

SESSION: 2023-24

School: School Of Computer Science & Information Technology	**Batch: 2022-2024**
Department: Department of Computer Applications	**Year: 2nd**
Course: Master of Computer Applications (MCA)	**Semester: IV**

S N	**Cate - gory**	**Course Code**	**Course Name**	**Periods**			**Credit s**	**Evaluation Scheme**					**Subjec t Total Marks**
								Theory			**Practical**		
				L	**T**	**P**		**AB Q**	**MS E**	**ES E**	**IP**	**EX P**	
4	PRO J	MCA-272	Major Project	0	0	36	18	-	-	-	-	100	100
			Total---->	**0**	**0**	**36**	**18**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
AEC C:	Ability Enhancement Compulsory course	ESE:	End Semester Examination
GE:	General English	IP:	Internal Practical
DSE:	Discipline Specific Elective Course	EXP:	External Practical
L:	Lecture	PROJ:	Project
T:	Tutorial		
P:	Practica l		

ELECTIVE-I	
CODE	**SUBJECT NAME**
MCA-122A	Advance Database management system
MCA-122B	Blockchain Technology
MCA-122C	Theory of computation

ELECTIVE-II	
CODE	**SUBJECT NAME**
MCA-217A	Compiler Design
MCA-217B	.net Framework using C#
MCA-217C	Mobile communication & Android Application Development

MCA-109	PROGRAMMING FOR PROBLEM SOLVING SKILLS	L T P	Cr
		3 0 0	3

OBJECTIVES

1. Design solutions to simple engineering problem by applying the basic programming principles of C language and basic mathematical knowledge.
2. Choose a suitable C-construct to develop C code for a given problem.
3. Recognize the bugs in the C program.
4. Apply the C-language syntax rules to correct the bugs in the C program.
5. Develop simple C programs to illustrate the applications of different data types such as arrays, pointers, functions.

PRE-REQUISITES

A Strongly recommended Computer fundamentals.

COURSE OUTCOMES

Students after undergoing this course will be able to:

CO1: Illustrate and explain the basic computer concepts and programming principles of C language.
CO2: Develop C programs to solve simple mathematical and decision-making problems.
CO3: Develop C programs to solve simple engineering problems using looping constructs.
CO4: Develop C programs to demonstrate the applications of derived data types such as arrays, pointers, strings and functions
CO5: Understanding and developing the c functions with their libraries.

Unit I

Introduction to Digital Computers: Representation of Algorithm, Flowcharts, Examples, Introduction to Programming: Importance of C, Basic Structure of C Programs, Programming Style, executing a C Program Constants, Variables, and Data Types: Introduction, CharacterSet, C Tokens, Keywords and Identifiers, Constants, Variables, Data Types, Declaration of Variables, Assigning Values to Variables, Defining Symbolic Constants. Managing Input and Output Operations: Reading a Character, Writing a Character, Formatted Input, Formatted Output. Operators and Expressions: Introduction, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operator, Bitwise Operators, Special Operators, Arithmetic Expressions, Evaluation of Expressions, Precedence of Arithmetic Operators, Type Conversions in Expressions, Operator Precedence and Associativity.

Unit II

Decision Making and Branching: Introduction, Decision Making with IF Statement, Simple IF Statement, the IF.....ELSE Statement, Nesting of IF....ELSE Statements, The ELSE IF Ladder, The Switch statement. (T1: Section 5.1-5.7) Decision Making and Looping: The WHILE Statement, The DO Statement, the FOR Statement, Jumps in LOOPS.

Unit III

Arrays: One-dimensional Arrays, Declaration of One-dimensional Arrays, Initialization of One-dimensional Arrays, Example programs- Linear search, Binary search, Bubble sort, Selection sort. Two-dimensional Arrays, Declaration of Two-dimensional Arrays, Initialization of Two-dimensional Arrays, Example programs – Matrix Multiplication, Transpose of a matrix.

Unit IV

Character Arrays and Strings: Declaring and Initializing String Variables, Reading Strings from Terminal, Writing Strings to Screen ,Arithmetic Operations on Characters, String-handling Functions (strlen(), strcpy(), strcmp(), strcat(), strrev()), Example Programs (with and without using built-in string functions), Two-dimensional character arrays. Pointers: Introduction, Declaring Pointer Variables, Initialization of Pointer variables, accessing a Variable through its Pointer, Pointer Expressions, Pointer Increments and Scale Factor, Pointers and 1-D Arrays.

Unit V

User-defined Functions: Elements of User-defined Functions, Definition of Functions, Return Values and their Types, Function Calls, Function Declaration, Category of Functions, No Arguments and no Return Values, Arguments but no Return values, Arguments with Return Values, No Arguments but Returns a Value, Passing Arrays to Functions. Recursion - Factorial of an integer, Xn , Finding n th Fibonacci numbers.

TEXT BOOKS:

1. Schaum's Outline of Programming with C by Byron Gottfried , McGraw-Hill
2. The C programming by Kernighan Brain W. and Ritchie Dennis M., Pearson Education .
3. Computer Basics and C Programming by V.Rajaraman , PHI Learning Pvt. Limited, 2015.
4. Computer Concepts and Programming in C, E Balaguruswami, McGraw Hill

Reference books:

1. Computer Science- A Structured Programming Approach Using C, by Behrouz A. Forouzan, Richard Gilberg, Thomson, Third Edition , Cengage Learning - 2007.
2. Let Us C By Yashwant P. Kanetkar.
3. Problem Solving and Program Design in C, by Jeri R. Hanly, Elliot B. Koffman, Pearson Addison-Wesley, 2006.

MCA-105	COMPUTER NETWORKS	L T P	Cr
		3 1 0	4

OBJECTIVES

To have a fundamental understanding of the design, performance and state of the art of wireless communication systems, Topics covered include state of the art wireless standards and research and thus changes substantially form one offering of this course to the next

PRE-REQUISITES

Knowledge of computers hardware and software

COURSE OUTCOMES

CO1: Understand computer network basics, network architecture,TCP/IP and OSI reference models.

CO2: Identify and understand various techniques and modes of transmission

CO3: Describe data link protocols, multi-channel access protocols and IEEE 802 standards for LAN

CO4: Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressing scheme

CO5: Discuss the elements and protocols of transport layer. Understand network security and define various protocols such as FTP, HTTP, Telnet, DNS

UNIT I

OVERVIEW OF DATA COMMUNICATION AND NETWORKING: Introduction; Data communications: components, data, direction of data flow, Protocols, Networks: type of connection, topology: Star, Bus, Ring, Mesh, Tree, categories of network: LAN, MAN, WAN: Internet: brief history, Layered architecture of networks, OSI reference model, Functions of each layer, services and protocols of each layer, TCP / IP reference model.

UNIT II

NETWORK LAYER AND TRANSPORT LAYER: Point -to Pont Networks, Introduction to Internet Protocol, IP Datagram, IP Addressing, routing, IP packet, IP address, IPv4 & IPv6. Congestion Control & Quality of services (QoS) –Congestion Control in TCP & Frame Relay Network; QOS; Flow Characteristics; Technique to improve Congestion Control; Scheduling; traffic shaping, Transport layer Protocols: TCP functions, segments and connections. UDP, TCP verses UDP.

UNIT III

WIDE AREA NETWORKS: Introduction to WAN, WAN technologies: SONET/SDH, ATM: ATM cell, layered architecture, ATM signaling, addressing and applications. Frame Relay Technology Overview and Standards. ISDN & B-ISDN: Technology Overview, Interfaces and Channels, Layered Protocol architecture and Frame Format.

UNIT IV

INTERNET SUITE OF PROTOCOLS: Application Layer: Introduction to application layer and protocols, WWW, HTTP, DNS. E-Mail and protocols: SMTP, IMAP and MIME.

File transfer protocols: FTP and TFTP. Network management protocol: SNMP. Voice over IP (VoIP).

UNIT V

Network Management and Security: Simple Network Management Protocol (SNNP). Data Encryption & Cryptographic techniques. Firewalls: types, architecture and applications, VLANs: architecture and applications, Internet Security protocol (IPsec).

TEXT BOOK

Tanenbaum Andrew S, ―Computer Networks‖, 4th Edition, Pearson Education/Prentice Hall of India, 2003.

REFERENCE BOOKS

1. Forouzan Behrouz A., ―Data Communications and Networking‖, Tata McGraw Hill 2006.
2. Stallings William, ―Data and Computer Com munication‖ , 5th Edition, Prentice Hall of India, 1997.
3. Fred H alsall, ―Data Communications, Computer Networks and Open Systems‖, 4th edition, Addison Wesley, Low Price Edition, 2000
4. Fitzgerald Jerry, ―Business Data Communications‖, Wiley, 2009.
5. Peterson Larry L. and Davie Bruce S., ―Computer Networks – A System Approach‖, 3rd Edition, Morgan Kaufmann, 2003.
6. Tittel E. D., ―Computer Networking‖, Tata McGraw Hill, 2002
7. Kurose James F. and Ross Keith W., ―Computer Networking: A Top-Down Approach Featuring the Internet‖, 2nd Edition, Pearson Education, 2003.
8. Keshav S., ―An Engineering Approach to Computer Networking‖, Addison-Wesley, 1997.
9. Comer D. E., ―Internetworking with TCP/IP‖, Volume 1, 3rd Edition, Prentice Hall of India, 1995.

MCA-111	DIGITAL LOGIC AND COMPUTER ARCHITECTURE	L	T/SDA	P	Credit
		3	0	0	3

OBJECTIVE

The objective of this course is to introduce the organization of a computer and its principal components, digital logic gates, combinational and logical circuits and micro operations

COURSE OUTCOMES

The students undergoing this course will be able to:
CO1: Understanding of Boolean algebra and Simplification of Boolean Functions
CO2: Understanding the digital logic gates
CO3: Understanding the Combinational Logic and Sequential Logic
CO4: Understanding of Microprocessor Architecture and Micro-operations
CO5: Understanding of CPU and Binary Arithmetic

UNIT I

Information Representation: Number Systems - Binary, Octal, Decimal, and Hexa-Decimal; Number Base Conversions; Binary Arithmetic; Complements: (r-1)'s Complement, r's Complement, Subtraction using Complements; Floating Number Fixed-point Representation, Floating-point Representation; Binary Codes for Decimal Digits: BCD Code, Excess-3 Code, 84- 2-1 Code, 2421 Code, Reflected Code; Error Detection Code; Character Representation – ASCII, EBCDIC.

UNIT II

Boolean Algebra, Logic Gates and simplification: Boolean Algebra-Basic Definitions, Postulate, Basic Theorems and Properties of Boolean Algebra; Boolean Functions, Canonical and Standard Forms: Minterms and Maxterms, SOP, POS Conversion Between Canonical Forms, Standard Form of a Boolean Function; Other Logical Operations; Digital Logic Gates, Implementation of Boolean Functions, Simplification using boolean Algebra and Karnaugh Maps (K-Map) Method.

UNIT III

Cobinational and Sequential Logic Circuit: Overview of Combinational Logic; Combinational Logic Design Procedure; Design of Some Standard Combinational Circuits: Half Adder, Full Adder, Half Subtractor, Full Subtractor, Code Conversion; Decimal Adder, BCD Adder, Magnitude Comparator, Decoders, Encoder, Multiplexers, De-multiplexer, Flip-Flops: RS Flip Flop, Clocked RS, JK Flip Flop, Master Slave JK Flip Flop, D Type Flip Flop, T Type Flip Flop, State Table, State Diagram, State Equations, Flip Flop Characteristic Tables; Flip Flop Excitation Tables; Design of Sequential Circuits.

UNIT IV

Register Transfer and Micro Operations: Register Transfer Language (RTL); Register Transfer; Bus Transfer; Memory Transfers; Arithmetic Microoperations; Logic Microoperations, List of Logic Microoperations, Addressing Modes, Data Transfer.

UNIT V

Central Processing Unit (CPU): Introduction; General Register Organization; Control Word; Stack Organization – Register Stack, Memory Stack, Reverse Polish Notation, Evaluation of Arithmetic Expression. Instruction Format – Three Address Instructions, Two Address Instructions, One Address Instructions, Zero Address Instructions. Parallel Processing; Pipelining – Arithmetic Pipeline, Instruction Pipeline

TEXT BOOKS

1. Mansaf Alam & Bashir Alam: Digital Logic Design.
2. PHI M. Morris Mano: Digital Logic and Computer Design.
3. Pearson M. Morris Mano: Computer System Architecture.

REFERENCE BOOKS

1. Pearson William Stalling: Computer Organization and Architecture.
2. Prentice Hall V. Rajaraman & T. Radhakrishnan: Computer Organization and Architecture.
3. PHI Donald D. Givone: Digital Principles and Design. McGraw Hill

MCA-113	DATA STRUCTURE AND ALGORITHMS	L T P	Cr
		3-0-0	0

COURSE OBJECTIVE: To relay the theoretical and practical fundamental knowledge of most basic data structure like array linked list, stack, queue, tree & graph To understand the implementation of these data structure ,to be familiar with basic techniques of algorithm analysis and analysis of the algorithms used for implementation of these data structure.

PRE-REQUISITES: Knowledge of basic computer programming.

COURSE OUTCOMES

CO1: Understand the concept of dynamic memory management, data types, algorithm, Big O notation.
CO2: Understand basic data structures such as arrays, linked list, stack and queue.
CO3: Describe the hash function and concepts of collision and its resolution methods.
CO4: Solve problem involving graph, trees and heaps.
CO5: Apply algorithm for solving problems like sorting, searching, insertion and deletion of data.

UNIT I: INTRODUCTION TO DATA STRUCTURES AND RUNNING TIME:
Definition of data structures and abstract data types; linear vs. non-linear data structure; primitive vs. non-primitive data structure; static and dynamic implementations; arrays, 1,2-dimensional arrays, insertion & deletion in 1-D array; examples and real life applications. Time complexity; Big Oh notation; running times; best case, worst case, average case; factors depends on running time; introduction to recursion.

UNIT II: STACKS AND QUEUES:
Stacks: definition, array based implementation of stacks, examples: infix, postfix, prefix representation; conversions, applications; definition of queues, circular queue; array based implementation of queues.

UNIT III: LINKED LISTS:
Lists; different type of linked Lists; implementation of singly linked list, linked list implementation of stacks and queues; implementation of circular linked list; implementation of doubly linked list, applications.

UNIT IV: TREES AND GRAPHS:
Definition of trees and binary trees; properties of binary trees and implementation; binary traversal pre-order, post-order, in-order traversal; binary search trees: searching, insertion & deletion. Definition of undirected and directed graphs; array based implementation of graphs; adjacency matrix; path matrix implementation; linked list representation of graphs; graph traversal: breadth first traversal, depth first traversal; implementations and applications.

UNIT V: SORTING AND SEARCHING ALGORITHMS:
Introduction, selection, insertions, bubble sort, efficiency of above algorithms; merge

sort, merging of sorted arrays and algorithms; quick sort algorithm analysis, heap sort, searching algorithms: straight sequential search, binary search (recursive & non–recursive algorithms).

TEXT BOOK

1. Langsam, Augentem M.J. and Tenenbaum A. M., ―Data Structures using C & C++‖, Prentice Hall of India, 2009.
2. R. S.Salariya, Data Structure and Algorithm, Khanna Publications.

REFERENCE BOOKS

1. Aho A. V., Hopcroft J. E. and Ullman T. D., ―Data Structures and Algorithms‖, Original Edition, Addison-Wesley, Low Priced Edition, 1983.
2. Horowitz Ellis and S ahni S artaj, ―Fundamentals of Data Structures‖, Addison-Wesley Pub, 1984.
3. Horowitz, S ahni and Rajasekaran, ―Fundamentals of Computer Algorithms‖ 2007.
4. Kruse Robert, ―Data Structures and Program Design in C‖, Prentice Hall of India, 1994
5. Lipschetz Jr. Seymour, ―Theory & Problems of Data Structures‖, S chaum ‘s Outline, Tata McGraw Hill
6. Weiss Mark Allen, ―Data Structures and Algorithms Analysis in C‖, Pearson Education, 2000
7. Corm en T . H . et al., ―Introduction to Algorithms‖, 2nd Edition, Prentice Hall of India, 2001.
8. Dasgupta Sanjay, Christos P. and Vazirani Umesh, ―Algorithms‖, Tata McGraw Hill, 2008

MCA-115	DATABASE MANAGEMENT SYSTEM	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To provide knowledge about various organizations and management information systems, keeping in view the aspects of shareability, availability, evolvability and integrity.

COURSE OUTCOMES

The students undergoing this course will be able:
CO1: To know the basics of database & its architecture
CO2: To aware about the existing data models, entities as well as constraints
CO3: To learn about the different anomalies of the data and ways to normalize it
CO4: To know about the storage of data in the files & organization of files
CO5: To learn about the transactions and its ways done on the database

UNIT I

INTRODUCTION: Basic Concepts: Data, Database and DBMS; Database vs. Traditional File System Approach; Three Schema Architecture of DBMS, Data Independence; Categories of Database Management Systems: Hierarchical, Network and Relational Database Systems.

UNIT II

DATA MODELING: Database Models: Introduction, Categories of Database Models: High-level or Conceptual Data Models, Representational or Implementation Data Models, Low-level or Physical Data Models, Object Data Models. Entity relationship (ER) Model: Basic Concepts and their representations – Entity, Entity Type and Entity Set, Attributes and Keys, Relationships, Relationship Types, and Structural Constraints, Weak Entity, Naming Conventions & Design Issues in ER Model. ER and EER Diagrams.

UNIT III

Relational Database Model: Structure of Relational Model; Domains, Attributes, Tuples, and Relations; Characteristics of Relations; Relational Constraints – Domain Constraints, Key Constraints, Entity Integrity, and Referential Integrity Constraints; Relational Database Schema; Relational Algebra Operations – Select, Project, Rename, Union, Intersection, Set Difference, Join, and Division Operations; Aggregate Functions and Groupings. Database design process; relational database design; relation schema; anomalies in a database; functional dependencies; 1NF, 2NF, 3NF, and BCNF. ; Decomposition of DBMS: Lossless and Lossy, Reduction of an E-R schema to Tables; Introduction to SQL; basic queries in SQL; advanced queries in SQL; functions in SQL; basic data retrieval; updates in SQLs, views in SQL. Relational Algebra: introduction, Selection and projection, set operations, renaming, Joins, Division, syntax, semantics. Operators, grouping and ungrouping, relational comparison. Calculus: Tuple relational calculus, Domain relational Calculus, calculus vs algebra, computational capabilities.

UNIT IV

Structured Query Language (SQL): Schema, Table and Domain Creation; Schema and Table Deletion; Table Modification; Insert, Delete, and Update Statements; SELECT- FROM-WHERE

Structure; Renaming Attributes; Nested Queries and Set Comparisons; EXISTS and UNIQUE Functions; Aggregate Functions; Creating and Updating Views. Introduction to PL/SQL.

UNIT V

Functional Dependencies and Normalization: Informal Design Guidelines for Relation Schemas; Functional Dependencies; Inference Rules for Functional Dependencies; Normalization using Functional Dependencies – First Normal Form (INF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF); Multi- Valued Dependencies and Fourth Normal Form (4NF); Join Dependencies and Fifth Normal Form (5NF); Relation Decomposition and Insufficiency of Normal Forms; Dependency Preserving and Lossless Join Decompositions; Null Values and Dangling Tuples. Transaction Management and Concurrency Control: Transaction Concept; Transaction State; Concurrent Executions; Serializability and Recoverability; Testing for Serializability. Concurrency Control – Lock-Based Protocols and Timestamp Based Protocols.

TEXT BOOK

1. Silberschatz A., Korth H. F. and Sudarshan S., “Database System Concepts”, 3rd edition, McGraw-Hill, International Edition, 1997.
2. S. Chand, Rajiv Chopra, “DBMS(A practical approach)”, Revised edition, Pearson.

REFERENCE BOOKS

1. Date C. J., “An Introduction to Database Systems”, 7th edition, Addison- Wesley, Low Priced Edition, 2000.
2. Elmasri R. and Navathe S. B., “Fundamentals of Database Systems”, 3rd edition, Addison-Wesley, Low Priced Edition, 2000.
3. A. Silberschatz, H. F. Korth, and S. Sudarshan: Database System Concepts. McGraw Hill
4. J. Casteel: ORACLE 9i Developer: PL/SQL Programming. Thomson
5. Ivan Bayross: SQL, PL/SQL The Programming Language of of Oracle. BPB

MCA-163	DATA STRUCTURES & ALGORITHMS LAB	L T P	Cr
		0 0 2	1

LIST OF EXPERIMENTS

ARRAY OPERATIONS

1. Write a program to insert an element at given position in linear array
2. Write a program to insert an element in sorted array.
3. Write a program to delete an element from given position in linear array
4. Perform following operations on matrices using functions only
   a) Addition b) Subtraction c) Multiplication d) Transpose

SEARCHING

5. Search an element in a linear array using linear search.
6. Using iteration and recursion concepts write programs for finding the element in the array using Binary Search Method

RECURSION

7. Write a program to compute factorial of given number using recursion
8. Write as program to solve Tower of Hanoi problem using recursion
9. Write a program to find power of given number using recursion

STACK & QUEUE

10. Write a program for static implementation of stack
11. Write a program for dynamic implementation of queue
12. Write a program for static implementation of circular queue
13. Write a program for dynamic implementation of queue
14. Write a program to evaluate a postfix operation

LINKED LIST

15. Create a linear linked list & perform operations such as insert, delete at end , at beg & reverse the link list
16. Create a circular linked list & perform search, insertion & delete operation
17. Create a doubly linked list & perform search, insertion & delete operation

TREE & GRAPH

18. Write program to implement binary search tree. (Insertion and Deletion in Binary Search Tree)
19. Write program to simulates the various tree traversal algorithms
20. Write program to simulate various graph traversing algorithms.

SORTING ALGORITHMS

21. Write program to implement Bubble, Insertion & selection sort.
22. Write program to implement quick sort
23. Write program to implement merge sort
24. Write a program to implement heap sort

TEXT BOOK

1. A.K. Sharma – Data structure Using C, 2nd edition pearson 2013

2. Langsam, Augentem M.J. and Tenenbaum A. M., ―Data Structures using C & C++‖, Prentice Hall of India, 2009.

REFERENCE BOOKS

1. R. S. Salaria -Data Structure Using C
2. Kruse Robert, ―Data Structures and Program Design in C‖, Prentice Hall of India, 1994
3. Lipschitz Jr. Seymour, ―Theory & Problems of Data Structures‖, Schaum‘s Outline, 2nd Edition, Tata McGraw Hill

MCA-159	Programming and Problem-Solving using C Lab	L T P	CR
		0 0 2	1

OBJECTIVES

1. Understand the basic principles of C programming language.
2. 2. Develop C programming skills.
3. 3. Develop debugging skills using Code Blocks IDE.

PRE-REQUISITES

While there are no requirements, a background in programming is strongly recommended.

LAB EXPERIMENTS:

1. WAP that calculates the Simple Interest and Compound Interest. The Principal, Amount, Rate of Interest and Time are entered through the keyboard.
2. WAP that accepts the marks of 5 subjects and finds the sum and percentage marks obtained by the student.
3. WAP to calculate the area and circumference of a circle.
4. WAP that accepts the temperature in Centigrade and converts into Fahrenheit using the formula C/5=(F32)/9.
5. WAP that swaps values of two variables using a third variable.
6. WAP that checks whether the two numbers entered by the user are equal or not.
7. WAP to find the greatest of three numbers.
8. WAP that finds whether a given number is even or odd.
9. WAP that tells whether a given year is a leap year or not.
10. WAP that accepts marks of five subjects and finds percentage and prints grades according to the following criteria: Between 90-100%--------------Print ‘A’ 80-90%-------------------------------Print ‘B’ 60-80%---------------------------Print ‘C’ Below 60%----------------------Print ‘D’
11. WAP that takes two operands and one operator from the user and perform the operation and prints the result by using Switch statement.
12. WAP to print the sum of all numbers up to a given number.
13. WAP to find the factorial of a given number.
14. WAP to print sum of even and odd numbers from 1 to N numbers.
15. WAP to print the Fibonacci series.
16. WAP to check whether the entered number is prime or not.
17. WAP to find the sum of digits of the entered number.
18. WAP to find the reverse of a number.
19. WAP to print Armstrong numbers from 1 to 100.
20. WAP to convert binary number into decimal number and vice versa.
21. WAP that simply takes elements of the array from the user and finds the sum of these elements.
22. WAP that inputs two arrays and saves sum of corresponding elements of these arrays in a third array and prints them.
23. WAP to find the minimum and maximum element of the array.
24. WAP to search an element in a array using Linear Search.
25. WAP to sort the elements of the array in ascending order using Bubble Sort technique.
26. WAP to add and multiply two matrices of order nxn.
27. WAP that finds the sum of diagonal elements of a mxn matrix.
28. WAP to implement strlen (), strcat (),strcpy () using the concept of Functions.
29. WAP to swap two elements using the concept of pointers.

30. WAP to compare the contents of two files and determine whether they are same or not.
31. WAP to check whether a given word exists in a file or not. If yes then find the number of times it occurs.

MCA-165	DATABASE MANAGEMENT SYSTEM LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

To provide knowledge about implementation of practical aspects of database i.e. creation of tables and applying queries using SQL queries

COURSE OUTCOMES

The students undergoing this course will be able:

CO1: To know the basics of structured query language

CO2: To aware about applying different queries on database structured in the form of tables

CO3: To learn about the different SQL queries performed using operators as well as constraints

CO4: To create views from created table to further organize the data

CO5: To learn about the basic operations of relational algebra

LIST OF EXPERIMENTS/EXERCISES

Basic

1. Introduction to SQL.
2. Write a query for:
   I. Creation of table.
   II. Insertion of data into table.
   III. Displaying the data of table.
   IV. Deletion of data from table.
   V. Updating the data in a table.
   VI. Modifying the structure of a table.

3. Write a query to implement: Key constraints: primary key constraints, foreign key constraints, not null constraints and unique constraints; use of check constraints.
4. Write a query to implement: Binary operations in Relational Algebra: Union, Intersection, Set Difference, Join, Cartesian product.
5. Write a query to implement: Grouping of data into tables and listing records in ascending order or descending order.
6. Write a query to implement: Creation of sequences and explain use of sequences.
7. Write a query to implement: Access permissions in SQL.

Moderate

1. Finding unique names of all salesmen, deletion of the structure of a table, use of delete command with conditions, updating records of a table with conditions, altering structure of a table and changing size of existing column in the table.
2. Write a query to implement: Arithmetic operators, logical operators and pattern matching operator.
3. Write a query to implement: Aggregate and mathematical functions: count, count(*), Avg, max, min, sum, lower, upper, power , sqrt.
4. Write a query to implement: Creating views from single and multiple tables, drop views and creating index on the table and drop them.
5. Create queries using Triggers and Procedures.

Advance

1. Consider the insurance database given below. The primary keys are made bold and the data types are specified.
   PERSON(driver_id:string , name:string , address:string)
   CAR(regno:string , model:string , year:int)
   ACCIDENT(report_number:int , accd_date:date , location:string)
   OWNS(driver_id:string , regno:string)
   PARTICIPATED(driver_id:string , regno:string , report_number:int , damage_amount:int)
   1)Create the above tables by properly specifying the primary keys and foreign keys.
   2)Enter at least five tuples for each relation.
   3)Demonstrate how you:
   a.Update the damage amount for the car with specific regno in the accident with report number 12 to 25000.
   b.Add a new accident to the database.
   4)Find the total number of people who owned cars that were involved in accidents in the year 2008.
   5)Find the number of accidents in which cars belonging to a specific model were involved.
2. Consider the following relations for an order processing database application in a company.
   CUSTOMER(custno:int , cname:string , city:string)
   ORDER(orderno:int , odate:date , custno:int , ord_amt:int)
   ORDER_ITEM(orderno:int , itemno:int , quantity:int)
   ITEM(itemno:int , unitprice:int)
   SHIPMENT(orderno:int , warehouseno:int , ship_date:date)
   WAREHOUSE(warehouseno:int , city:string)
   1)Create the above tables by properly specifying the primary keys and foreign keys.
   2)Enter at least five tuples for each relation.
   3)Produce a listing: custname , No_of_orders , Avg_order_amount , where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.
   4)List the orderno for orders that were shipped from all the warehouses that the company has in a specific city.
   5)Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER_ITEM table that contains this particular item.
3. Consider the following database of student enrollment in courses and books adopted for that course.
   STUDENT(regno:string , name:string , major:string , bdate:date)
   COURSE(courseno:int , cname:string , dept:string)
   ENROLL(regno:string , courseno:int , sem:int , marks:int)
   BOOK_ADOPTION(courseno:int , sem:int , book_isbn:int)
   TEXT(book_isbn:int , book_title:string , publisher:string , author:string)
   1)Create the above tables by properly specifying the primary keys and foreign keys.
   2)Enter atleast five tuples for each relation.
   3)Demonstrate how you add a new text book to the database and make this book to be adopted by some department.
   4)Produce a list of text books (includes courseno , book_isbn , book_title) in the alphabetical order for courses offered by the 'CS' department that use more than two

books.
5)List any department that has all its books published by a specific publisher.

4. The following are maintained by a book dealer.
   AUTHOR(author_id:int , name:string , city:string , country:string)
   PUBLISHER(publisher_id:int , name:string , city:string , country:string)
   CATALOG(book_id:int , title:string , author_id:int , publisher_id:int , category_id:int , year:int , price:int)
   CATEGORY(category_id:int , description:string)
   ORDER_DETAILS(order_no:int , book_id:int , quantity:int)
   1)Create the above tables by properly specifying the primary keys and foreign keys.
   2)Enter at least five tuples for each relation.
   3)Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
   4)Find the author of the book that has maximum sales.
   5)Demonstrate how you increase the price of books published by a specific publisher by 10%.
5. Consider the following database for a banking enterprise.
   BRANCH(branch_name:string , branch_city:string , assets:real)
   ACCOUNT(accno:int , branch_name:string , balance:real)
   DEPOSITOR(customer_name:string , accno:int)
   CUSTOMER(customer_name:string , customer_street:string , customer_city:string)
   LOAN(loan_number:int , branch_name:string , amount:real)
   BORROWER(customer_name:string , loan_number:int)
   1)Create the above tables by properly specifying the primary keys and foreign keys.
   2)Enter at least five tuples for each relation.
   3)Find all the customers who have at least two accounts at the main branch.
   4)Find all the customers who have an account at all the branches located in a specific city.
   5)Demonstrate how you delete all account tuples at every branch located in a specific city.

Project

1. Create a Database for a college which implements all keys, views, procedures, and triggers in it.

MCA-112	SOFTWARE ENGINEERING AND TESTING	L-T-P	Cr
		3-1-0	4

OBJECTIVE

The objective of the courses is to make the learner efficiently work as a software engineer. The students should be well acquainted with all the phases of the Software Development Life Cycle. The learner should be able to apply the concepts learned for doing research

PRE-REQUISITES

Knowledge of fundamentals of software and testing basics.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: To know about the basic concepts of software development

CO2: To learn about the various techniques of software design as well as create a design

CO3: To create different test cases in different situations

CO4: To use different tools for building test reports, risks as well as test cases

CO5: To automate the testing process

UNIT I

SOFTWARE ENGINEERING MODELS AND REQUIREMENT ANALYSIS: Evolution of Software Engineering, SDLC, Software development models, Software process technology, Requirement Gathering phase, Methods of Requirement gathering, Creation of SRS

UNIT II

SOFTWARE DESIGN: Data design, Architectural design, Interface design, Modular design, Creation of DFD, Levels of DFD, Types of Software Design, Steps to create Software design, Issues of Software design

UNIT III

SOFTWARE TESTING CONCEPTS: Need of Testing, Debugging, Testing Techniques, Functional testing and its types, Non functional testing and its types, White Box Testing, Black Box Testing, Structural versus Functional Technique, Verification versus Validation, Static versus Dynamic Testing,

Test Administration Test Planning, Customization of the Test Process, Budgeting, Scheduling

Create the Test Plan, Prerequisites to test planning, Understand the Characteristics of the Software Being Developed, Build the Test Plan, Write the Test Plan

UNIT IV

TEST CASES: Test case Design, Building test cases, Test data mining, Test execution, Test Reporting, Defect Management, Test Coverage – Traceability matrix, Test Metrics – Guidelines and usage, Test reporting: Guidelines for writing test reports

UNIT V

TEST TOOLS: Test Tools used to Build Test Reports, Managing Change Software Configuration Management, Change Management, Risks – Risk Analysis and Management with examples

User Acceptance testing – in detail explanation with details.

Case Study: How to test web, stand alone and database applications – with examples.

TEXT BOOKS

1. A Practitioner's Guide to Software Test Design, Lee Copeland, 2003
2. The Art of Software Testing, 2nd edition, Glenford Myers, et. el., 2004

REFERENCE BOOKS

1. Software Testing Techniques, 2nd edition, Boris Beizer, 1990
2. How to Break Software: A Practical Guide to Testing, James Whittaker, 2002
3. Testing Object-Oriented Systems: Models, Patterns, and Tools, Robert V. Binder, 1999

MCA-114	OBJECT ORIENTED PROGRAMMING USING C++	L-T-P	Cr
		4-0-0	4

OBJECTIVE

To build programming logic and thereby developing skills in problem solving using C++ programming language; Introduce the concept of object orientation and on how to handle data in different forms; Emphasize the concepts and constructs rather than on language features.

PRE-REQUISITE:

C is a basic programming language whereas C++ is pure object oriented language. An added advantage of learning the basics of C first is that every part of what you are probably going to learn in C++ would already be covered by you while learning C.

COURSE OUTCOMES

CO1: Describe the procedural and object-oriented paradigm with concepts of streams, classes, functions, data and objects.

CO2: Understand dynamic memory management techniques using pointers, constructors, destructors, etc

CO3: Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.

CO4: Demonstrate the use of various OOPs concepts with the help of programs.

CO5: Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.

UNIT I

OO CONCEPTS: Programming Paradigms: Unstructured Programming, Structured Programming, Object Oriented Programming; ADT; Class; Object; Message; Encapsulation; Polymorphism; Inheritance; Pros and Cons of Object-oriented Methodology; cin and cout Objects.

UNIT II

CLASSES AND OBJECTS: Classes; Friend Functions: Benefits and Restrictions, Friends Classes; Inline Functions; Constructor, Parameterized Constructor; Destructor and its usages; Static Data Member and Static Member Functions; Creating Object; Passing and Returning Object(s) to/from a Function; Object Assignment; Nested and Local Classes; Arrays of Objects; Pointer to Objects; this Pointer, Pointer to Derived Type; References; Reference vs Pointer; Reference Parameters; Dynamic Memory Allocation.

UNIT III

FUNCTION AND OPERATOR OVERLOADING: Function overloading: Rules, Overloading Constructors, Copy Constructors; Default Function Arguments vs. Function Overloading. Operator Overloading: Operators that cannot be Overloaded, Overloading Operators using Member Function and Friends Functions, Overloading different operators including prefix and postfix form of ++ and – operators.

UNIT IV

INHERITANCE & VIRTUAL FUNCTION: Inheritance: Types of Inheritances, Base-Class Access Control, Protected Members, Protected Base-class Inheritance, Multiple Inheritance and problems, Solution to Multiple Inheritance Problem, Passing Parameters to Base Class Constructors; Virtual functions: Introduction, Calling a Virtual Function using Base Class Reference, Pure Virtual Function, Abstract Class.

UNIT V

GENERIC FUNCTION, EXCEPTION AND FILE HANDLING: Generic Functions: Benefits, Functions with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a generic function, Restriction, Generic Classes. Exception Handling, user defined Exception. C++ Streams; C++ File Handling: Opening/Closing a File, Reading /Writing a Text File, Random Access, Reading /Writing Object to a File.

TEXTBOOK

Balagurusamy, E., "Object Oriented Programming with C++", Prentice Hall of India, 2008.

Stroustrup, B., "The C++ Programming Language (4th Edition)", Addison-Wesley ISBN 978-0321563842. May 2013.

Yashavant, Y., "Let Us C++", BPB Publication, March 2020.

REFERENCE BOOKS

1. Kamthane, "Object Oriented Programming with ANSI and Turbo C++", Pearson Education
2. Lafore, Robert, "Object Oriented Programming in Turbo C++", The WAITE Group Press, 1994
3. Schildt, Herbert "C++: The Complete Reference", Tata McGraw Hill, 3rd Ed, 2008
4. Bhave, "Object Oriented Programming with C++", Pearson Education
5. Herbert Schildt: The Complete Reference C++. McGraw Hill
6. H.M. Deitel & P.J. Deitel: C++ How to Program. PHI
7. A. N. Kamthane: Object Oriented Programming with ANSI and TURBO C++. Pearson

MCA-116	OPERATING SYSTEM AND SHELL PROGRAMMING	L-T-P	Cr
		3-1-0	4

OBJECTIVE

To provide the knowledge of internals, different types and purpose of operating systems and aims to introduce open source operating systems.

PRE-REQUISITES

Knowledge of the fundamentals of Programming..

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: To understand the basics of operating systems like kernel, shell, types and views of operating systems

CO2: To describe the various CPU scheduling algorithms.

CO3: To explain various process synchronization techniques and concept of deadlocks.

CO4: To get an idea of shell scripts and types of shells.

CO5: To build shell programs and get an idea of AWK.

UNIT I

INTRODUCTION: Introduction to operating system concepts (including multitasking, multiprogramming, multi user, multithreading, etc)., types of operating systems: batch operating system, time-sharing systems, distributed OS, network OS, real time OS, embedded and smart card OS, various operating system services, architecture, system programs and calls.

UNIT II

PROCESS MANAGEMENT: Process concept, Life cycle and implementation of process, Thread usage and implementation in user space and in kernel, process scheduling, operation on processes, CPU scheduling, scheduling criteria, scheduling algorithms -First Come First Serve (FCFS), Shortest-Job-First (SJF), priority scheduling, Round Robin (RR), multilevel feedback queue scheduling.

UNIT III

DEADLOCKS AND PROCESS SYNCHRONIZATION: Deadlock characteristics, prevention, avoidance using banker's algorithm, detection and recovery; Critical section problems, mutual exclusion with busy waiting, Process synchronization, semaphores: binary and counting semaphores, Classical IPC problems: dining philosophers' problem, readers-writers problem.

UNIT IV

INTRODUCTION TO SHELL PROGRAMMING: Shell Variables, Environment Variables, Shell Scripts, read, Using Command Line Arguments, exit and exit status of command, The Logical Operators, The if Conditional, using test and [] to Evaluate Expression, The case Conditional, expr, while: looping, for: looping with a list, set and shift

UNIT V

AWK AND ADVANCED SHELL PROGRAMMING: Simple AWK Filtering, Splitting a Line into Fields, printf, the Logical and Relational Operators, Number Processing, Variables, The –f option, BEGIN and END positional Parameters, getline, Built-invariables, Arrays, Functions, Interface with the Shell, Control Flow. The sh command, export Command, Conditional Parameter Substitution, Merging Streams, Shell Functions, eval, Exec Statement and Examples

TEXT BOOKS

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 8th Edition, 2014, ISBN10: 0133805913 • ISBN13: 9780133805918
2. Sumitabha Das: UNIX Concepts and Applications, 4th Edition, Tata McGraw Hill, 2006.

REFERENCE BOOKS

1. Harvey M. Deitel, Operating Systems, Prentice Hall, 3rd Edition,2003, ISBN10: 0131828274|ISBN13: 9780131828278
2. Stallings William, “Operating Systems Internals and Design Principles”, 4th edition, Prentice-Hall, 2001.
3. UNIX: The Complete Reference: Kenneth Roson et al, Osborne/McGraw Hill, 2000.

MCA-118	ARTIFICIAL INTELLIGENCE	L T P	Cr
		3 0 0	3

COURSE OBJECTIVE:

To introduce about artificial intelligence approaches to problem solving, various issues involved and application areas

PRE-REQUISITES: Knowledge of discrete mathematics and data structures.

COURSE OUTCOMES

The students undergoing this course will be able:

CO1: To know the basics of artificial intelligence

CO2: To learn the different searching techniques of artificial intelligence

CO3: To learn about the representation of different information to produce a system

CO4: To know about different planning and learning.

CO5: To aware about the uncertainties

UNIT I : INTRODUCTION TO AI AND SEARCH TECHNIQUES:

Foundation and history of AI; data, information and knowledge; AI problems and techniques – AI programming languages, problem space representation with examples; blind search strategies, breadth first search, depth first search, heuristic search techniques: hill climbing: best first search, A * algorithm AO* algorithm, Minimax search procedure for Game Playing.

UNIT II : KNOWLEDGE REPRESENTATION ISSUES AND TECHNIQUES:

Predicate logic; representing knowledge using rules. Semantic nets, partitioned nets, parallel implementation of semantic nets; frames, forward and backward chaining; frame based systems.

UNIT III : REASONING UNDERUNCERTAINITY:

Reasoning under uncertainty, non monotonic reasoning; Review of probability; Baye's probabilistic interferences and Dumpster Shafer theory; statistical reasoning, fuzzy reasoning.

UNIT IV : PLANNING & LEARNING:

Goal stack planning; non linear planning, hierarchical planning representation for planning; partial order planning algorithm. Basic concepts of Learning ; rote learning, learning by taking advices, learning by problem solving, learning from examples, discovery as learning, learning by analogy; explanation based learning; neural nets; genetic algorithms.

UNIT V: EXPERT SYSTEM AND APPLICATIONS OF ARTIFICIAL INTELLIGENCE:

Expert systems: rule based systems architecture: Principles of natural language processing: knowledge acquisition concepts; AI application to robotics, and current trends in intelligent systems; parallel and distributed AI.

TEXT BOOK

1. Rich Elaine and Knight Kevin, ―Artificial Intellig ence‖ 3rd Edition, Tata McGraw Hill, 1991

REFERENCE BOOKS

1. Nilson Nils J., —Artificial Intelligence‖, McGraw-Hill, New York 1971
2. Russell Stuart and Norvig Peter, —Artificial Intelligence: A Modern Approach‖, Prentice Hall of India, 1998
3. Negnevitsky, —Artificial Intelligence: A Guide to Intelligent System ‖, Pearson Education, 2004.
4. Patterson O. W., —Introduction to Artificial Intelligence & Expert Systems‖, Prentice Hall of India, 1996.
5. Winston Patrick Henry, —Artificial Intellig ence‖, 3rd Edition, Addition Wesley, 1992
6. Clockson & Mellish, —Programming PROLOG‖, 3rd Edition, Narosa Publications, 2002.

MCA-120	STATISTICAL AND NUMERICAL COMPUTING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

Students will be able to understand basic theoretical and applied principles of statistics needed to enter the job force. Students will be able to communicate key statistical concepts to non-statisticians. Students will gain proficiency in using statistical software for data analysis.

PRE-REQUISITE:

This subject requires the basic knowledge of Linear Algebra and Computing. An added advantage of learning the basics of R programming first will be beneficial, as R is a language and an environment that is used for statistical computing and graphics.

COURSE OUTCOMES

CO1: To apply discrete and continuous probability distributions to various business problems.

CO2: To perform Test of Hypothesis as well as calculate confidence interval and understand the concept of p-values.

CO3: To learn non-parametric and parametric tests in order to analysis and interpret the data

CO4: To compute Bivariate and Multivariate Correlation, Regression and perform ANOVA, ANCOVA and MANOVA

UNIT I

OVERVIEW OF R: R data types and objects, reading and writing data; Control structures, functions, scoping rules, dates and times; Loop functions, debugging tools; Simulation, code profiling.

UNIT II

SOLUTION OF EQUATIONS AND SYSTEM OF SIMULTANEOUS EQUATIONS: Solution of Algebraic and Transcendental Equations using Bisection, Regula False, and Newton Raphson Methods, Gauss Elimination, Gauss Seidel, and Jacobi Methods.

UNIT III

INTERPOLATION, NUMERICAL DIFFERENTIATION AND INTEGRATION, AND DIFFERENTIAL EQUATIONS: Interpolation using Lagrange, and Newton’s methods, Extrapolation, Least Square Fitting, Numerical Integration using Trapezoidal, and Simpson’s Rules, Numerical Solution of Ordinary Differential Equations using Euler’s and Range-Kutta Methods.

UNIT IV

STATISTICS: Population, Sample, Sample Collection Methods, Data Representations and Classification, Central Tendency and Dispersion: Mean, Median and Mode, Quartiles and Percentiles, Measures of Dispersion: Range, Variance, Standard Deviation, and Coefficient of Variation. Skewness, and Kurtosis.

UNIT V

Probability and Hypothesis Testing: Sample Space, Events, Equally Likely Events, Probability, Independent Events, Addition and Multiplication Rules, Conditional Probability, Probability Distributions – Normal, Binomial, and Poison Distributions; Hypothesis Testing: Correlation using Karl Pearson and Spearman Rank Methods; Linear Regression; t-Test, Chi-Square Test, Analysis of Variance (ANOVA), Analysis of covariance (ANCOVA), Multivariate Analysis of Variance (MANOVA).

TEXTBOOK

1. M. K. Jain, S. R. K. Iyengav, and R. K. Jain: Numerical Methods for Scientific and Engineering Computation. New Age
2. Andy Field, J. Miles, and Z. Field: Discovering Statistics Using R. SAGE

REFERENCE BOOKS

S.C. Chapra & R.P.Canale: Numerical Methods for Engineering. TMH
V. Rajaraman: Computer oriented numerical methods. PHI
A. S. Grewal: Higher Engineering Mathematics. Khanna

MCA-164	OBJECT ORIENTED PROGRAMMING USING C++ LAB	L-T-P	Cr
		0-0-2	1

COURSE OUTCOMES

CO1: Develop solutions for a range of problems using objects and classes.
CO2: Programs to demonstrate the implementation of constructors, destructors and operator overloading.
CO3: Apply fundamental algorithmic problems including type casting, inheritance, and polymorphism.
CO4: Understand generic programming, templates, file handling.

LIST OF EXPERIMENTS/EXERCISES

BASIC

1. Write a program to perform basic arithmetic operations
2. Write a program to implement if statement
3. Write a program to implement if else ladder
4. Write a program to implement Switch Statement
5. Write a program to implement for loop
6. Write a program to implement nested for loop
7. Write a program to implement while loop
8. Write a program to implement nested while loop
9. Write a program to implement do-while loop
10. Write a program to implement do-while nested loop
11. Write a program to implement break statement
12. Write a program to implement continue statement

MODERATE

13. Write a program to implement function overloading
14. Write a program to implement the concept of class and object
15. Write a program to implement the concept of static data member
16. Write a program using the concept of constructor & destructor
17. Write a program to implement operator overloading
18. Write a program to implement access modifiers

19. Write a program to Implement single inheritance
20. Write a program to Implement Multiple inheritance

ADVANCED

21. Write a program to implement the concept of Friend Function

22. Write a program to Implement Virtual Function

23. Write a program to create, read & write sequential file

24. Write a program to create, read & write random access file

25. Write a program to implement function template

26. Write a program to overload function template

27. Write a program to implement class template

28. Write a program to implement exception handling

Projects to Enhance Your C++ skills

1. Login and Registration System
2. Car Rental System
3. Student Report Management System
4. Bookshop inventory system
5. Tic Tac Toe Game C++ Projects.

REFERENCE BOOK

1. Barkakati, Nabajyoti, “Object Oriented Programming in C++”, Prentice Hall of India, 2001.
2. Schildt, Herbert, “C++: The Complete Reference”, Tata McGraw Hill, 4th Edition, 2003

MCA-168	ARTIFICIAL INTELLIGENCE LAB	L T P	Cr
		0-0-2	1

LIST OF EXPERIMENTS

1. Study of Python programming language.
2. Write a program to find out route distance between two cities using python.
3. Write a program to implement Tower of Hanoi using python.
4. Write a program to calculate factorial of a number using python..
5. Write a program to implement Hardware simulation using Gates using python.
6. Write a program to implement family relationship using python.
7. Write a program to implement logon with recursion using python.
8. Write a program to print the list of customer having different colored cars with price and model available using python..
9. Write a program to implement water jug problem using python.
10. Write a program to implement Breadth First Search using python.
11. Write a program to implement Depth First Search using python.
12. Write a program to implement five House logic puzzle problem using python.
13. Write a program to analyze Grammar of sentences using python..
14. Write a program to solve 8-Queens problem using python.
15. Write a program to solve Monkey Banana problem using python.

MCA-170	STATISTICAL & NUMERICAL COMPUTING LAB	L-T-P	Cr
		0-0-2	1

COURSE OUTCOMES

CO1: Data manipulation, plot the graphs and charts with the help of computing features in R Programming.

CO2: The given data Interpretation with different distribution functions

CO3. the relevance and importance of the theory in solving practical problems in the real world

CO4: To learn non-parametric test such as the Chi-Square test for Independence as well as Goodness of Fit.

CO5: To compute Bivariate and Multivariate Regression and Correlation and perform ANOVA and F-test.

LIST OF EXPERIMENTS

INTRODUCTION TO COMPUTING

a) Installation of R
b) The basics of R syntax, workspace
c) Matrices and lists
d) Sub setting
e) System-defined functions; the help system
f) Errors and warnings; coherence of the workspace

GETTING USED TO R: DESCRIBING DATA

a) Viewing and Manipulating Data
b) Plotting Data
c) Reading the Data from console, file (.csv) local disk and Web
d) Working with larger datasets

VISUALIZING DATA

a) Tables, charts and plots.
b) Visualizing Measures of Central Tendency, Variation, and Shape.
c) Box plots, Pareto diagrams.
d) Find the mean, media, standard deviation and quantiles of a set of observations.
e) Note: Experiment with real as well as artificial data sets.

PROBABILITY DISTRIBUTIONS

a) Random number generation Distributions, the practice of simulation
b) Generate and Visualize Discrete and continuous distributions using the statistical environment.
c) Demonstration of CDF and PDF uniform and normal, binomial Poisson distributions.
d) Generate artificial data using and explore various distribution and its properties. Various parameter changes may be studied.

EXPLORATORY DATA ANALYSIS

Demonstrate Range, summary, mean, variance, median, stdev, histogram, box plot, scatterplot

APPLY-TYPE FUNCTIONS

a) Defining user defined classes and operations, Models and methods in R

b) Customizing the user's environment
c) Conditional statements
d) Loops and iterations

TESTS OF HYPOTHESES

a) Perform tests of hypotheses about the mean when the variance is known.
b) Compute the p-value.
c) Explore the connection between the critical region, the test statistic, and the p-value
a) Demonstrate Statistical functions in R
b) Statistical inference, contingency tables, chi-square goodness of fit, regression, generalized linear models, advanced modelling methods

CORRELATION

a) How to calculate the correlation between parametric and non-parametric data.
b) How to make scatter plots.
c) Use the scatter plot to investigate the relationship between two variables

T-TEST

a) Single Sample T-Test
b) Independent Sample T-Test
c) Paired Sample T-Test

Chi Square Test

a) Test for Independence
b) Goodness of Fit

ESTIMATING A LINEAR RELATIONSHIP

Demonstration on a Statistical Model for a Linear Relationship (Linear Regression)
a) Least Squares Estimates
c) Scrutinizing the Residuals

ANALYSIS of VARIANCE, ANALYSIS of CO-VARIENACE, MULTIVARIATE ANALYSIS of VARIANCE

a) Understating the concept, performing the test, and interpreting the results of ANOVA, ANCOVA and MANOVA

Reference Books:

1. Maria Dolores Ugarte , Ana F. Militino , Alan T. Arnholt "Probability and Statistics with R", on, CRC Press, 2 nd Edition 2016.
2. P. Dalgaard. "Introductory Statistics with R", 2nd Edition, Springer, 2008.

Web References:

1. http://nptel.ac.in/courses/110106064/

MCA-209	OPERATIONAL RESEARCH & OPTIMIZATION	L-T-P	Cr
		4-0-0	4

OBJECTIVE

Operations research (OR) has many applications in science, engineering, economics, and industry and thus the ability to solve OR problems are crucial for both researchers and practitioners. Being able to solve the real-life problems and obtaining the right solution requires understanding and modeling the problem correctly and applying appropriate optimization tools and skills to solve the mathematical model. The goal of this course is to teach you to formulate, analyze, and solve mathematical models that represent real-world problems. We will also discuss how to use EXCEL and LINDO for solving optimization problems. In particular, we will cover linear programming, network flow problems, integer programs, nonlinear programs, dynamic programming and queueing models.

COURSE OUTCOMES

The students undergoing this course will be able:
CO1: Formulate a real-world problem as a mathematical programming model and Implement and solve the model in EXCEL and LINDO
CO2: Understand the relationship between a linear program and its dual, including strong duality and complementary slackness
CO3: Solve network models like the shortest path, minimum spanning tree, and maximum flow problems
CO4: Understand how to model and solve problems using dynamic programming
CO5: Learn optimality conditions for single- and multiple-variable unconstrained and constrained non-linear optimization problems, and corresponding solution methodologies

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 max-mini, mini-max 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
Queueing 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,pseudo random numbers, multiplication congruential algorithm, inverse transformation method, basic ideas of Monte-Carlo simulation.

TEXT BOOK

1. Kanti Swarup, Gupta.P.K. & Man Mohan, operations Research, S.Chand & Sons.
2. Frank S. Budnick, Dennis Mcleavey and Richard Mojena, Principles of Operations Research for Management. All India Traveler Book seller, Delhi.

REFERENCE BOOKS

1. Joseph.G.Ecker & Michael Kupper Schimd, Introduction to operations Research, John Wiley & Sons,
2. Gillet.B.E., Introduction to Operations Research - A Computer oriented algorithmic approach, McGraw Hill

MCA-211	DIGITAL IMAGE PROCESSING	L	T/SDA	P	Credit
		3	1	0	4

OBJECTIVE

To introduce the concepts of image processing and basic analytical methods to be used in image processing. To familiarize students with image enhancement and restoration techniques, to explain different image compression techniques. To introduce segmentation and morphological processing techniques.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Understand the need for image transforms different types of image transforms and their properties.

CO2: learn different techniques employed for the enhancement of images, different causes for image degradation and various image restoration techniques.

CO3: Understand the need for image compression and to learn spatial and frequency domain techniques of image compression

CO4: Learn different Image Segmentation techniques.

CO5: Learn Image classification using Deep learning

UNIT I

Introduction and Spatial filtering: Components of a DIP system, Elements of visual perception, Light and Electromagnetic Spectrum, Image sensing, acquisition, Sampling and Quantization, Spatial and Intensity resolution, Basic relationships between pixels, Basic mathematical tools used in DIP, Intensity transformation functions, Histogram processing, Low pass and High pass Spatial Filters, 2D and 3D Convolutions

UNIT II

Frequency domain filtering and Image Restoration: 1-D and 2-D Discrete Fourier Transform (DFT) and IDFT, Low pass, High pass, selective Frequency Domain filters, Fast Fourier Transform (FFT), Model of Image restoration and Reconstruction, Noise models, Restoration in presence of noise only-Spatial filtering, Max, Min, Mean, Median filters, Periodic Noise reduction using frequency domain filtering, Notch filter.

UNIT III

Color Image Processing: Color Models: RGB, HSI, CMYK, YCbCr, Full Color Image Processing, Color image smoothing, sharpening, Using color in image segmentation, Intro to wavelet and other image transform, Introduction to Morphological Image Processing.

UNIT IV

Image Compression: Relative Data Redundancy (RDR), Compression Ratio, Coding Redundancy, Fidelity criteria, Lossless v/s Lossy Image Compression, Image compression Model, Huffman coding, Run-length coding, Bit plane coding, Image formats and standards, KLT, JPEG-steps etc., , Watermarking

UNIT V

Image Segmentation and Classification: Point, Line, and Edge detection, thresholding, Edge and Region based segmentation, Region Segmentation using Clustering, Feature extraction: Boundary and Region feature descriptors, Principal components, whole image features etc, Image datasets, Image Classifiers using Neural Networks, Deep learning, Deep Convolutional Neural Networks etc.

TEXT BOOKS

1. R. C. Gonzalez and R. E. Woods: Digital Image Processing.
2. Pearson R. C. Gonzalez, R. E. Woods, and S. Eddins: Digital Image Processing using MATLAB.
3. Mc Graw Hill M. Sonka, V. Hlavac, and R. Boyle: Image Processing, Analysis, and Machine Vision.
4. Vikas Publishing House Ashwin Pajankar: Python 3 Image Processing. Bpb

REFERENCE BOOKS

5. “Fundamentals of Digital Image Processing” by Sanjay Sharma
6. “Fundamentals of Digital Image Processing” by ANNADURAI
7. “Digital Image Processing” by Gonzalez and Woods
8. “Digital Image Processing” by William K Pratt

MCA-205	PYTHON PROGRAMMING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To build programming logic and thereby developing skills in problem solving using Python programming language; To be able to do testing and debugging of code written in Python Emphasize the concepts and constructs rather than on language features.

PRE-REQUISITES

The students are expected to have basic knowledge of programming and DBMS.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: To learn and understand Python programming basics, looping, control statements and string manipulations.

CO2: To learn basic concepts of list, tuple and dictionary data structures.

CO3: To learn and know the concepts of Structure and Functions.

CO4: To learn and know the concepts of file handling, exception handling.

CO5: To learn how to analyze and visualize the data.

UNIT I:

Introduction to Python: History, Features & Benefits of Python, Structure of a Python Program, Identifiers and Keywords, Concept of Variable, Memory Allocation for Variable, Data Types in Python, Conversion Functions, Operators (Arithmetic Operator, Relational, Logical or Boolean Operator, Assignment, Bitwise Operator, Membership Operator).Input and Output Function, Control Statements (Looping- while Loop, for Loop, Loop Control, Conditional Statement-if...else), Difference between break, continue and pass.

UNIT II:

Data Structures & Function: String, Lists, Tuples, Sets, Dictionary Data Structure, Built-in Library Function, Method and Operation on these Data Structure. Defining Function, Type of Function Arguments (Required Arguments, Keyword Arguments, Default Arguments, Variable-Length Arguments), Scope of a Variable, Global Vs Local Variable, Python Modules & Packages, Import Statement, dir(), globals(), locals() and reload() Functions.

UNIT III:

Python Object Oriented Programming: Features of Object Oriented Programming, Creating Classes, _init_() Method, Creating Instance Object, Class Attributes, Access Specifiers in Python, Instance Method Vs Class Method. Inheritance & Polymorphism, Overriding and Overloading Methods.

UNIT IV:

Python File Handling, Exception Handling: Opening & Closing Files, File Access Modes, File Object Attributes, Reading and Writing Files, Manipulating File Pointer using seek and tell. Programming using File Operations. Exception Handling in Python.

UNIT V:

Data Analysis: Dataset generation, Importing Dataset: Importing and Exporting Data, Basic Insights from Datasets, Cleaning and Preparing the Data: Identify and Handle Missing Values. Numpy and Scipy Package, Understanding and creating N-dimensional arrays, basic indexing and slicing, Boolean indexing, fancy indexing, Universal functions, Data processing using arrays, File input and output with arrays.

Data Visualization: Basic Visualization Tools, Specialized Visualization Tools, Creating and Plotting Maps.

TEXT BOOKS

1. John V Guttag. “Introduction to Computation and Programming Using Python”, Prentice Hall of India.
2. T. Budd, Exploring Python, TMH, 1st Ed, 2011

REFERENCE BOOKS

1. Python Tutorial/Documentation www.python.or 2010
2. Allen Downey, Jeffrey Elkner, Chris Meyers ,How to think like a computer scientist :Learning with Python,Freelyavailableonline.2012
3. http://docs.python.org/3/tutorial/index.html
4. http://interactivepython.org/courselib/static/pythonds

MCA-213	SOFT COMPUTING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To introduce the soft computing concepts and techniques and to foster their abilities in designing appropriate technique for a given scenario. To implement soft computing based solutions for real world problems. T give students knowledge about non-traditional techniques and fundamentals of artificial neural networks, fuzzy logic and genetic algorithms. To provide students hands-on experience on MATLAB to implement various strategies

PRE-REQUISITES

Knowledge of Mathematics.

COURSE OUTCOMES

The students undergoing this course will be able:

CO-1 Identify and describe soft computing techniques and their roles in building intelligent machines.

CO-2 Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems.

CO-3 Apply genetic algorithms to combinatorial optimization problems.

CO-4 Evaluate and compare solutions by various soft computing approaches for a given problem.

CO-5 Use various tools to solve soft computing problems.

UNIT I

INTRODUCTION TO SOFT COMPUTING AND NEURAL NETWORKS: Evolution of Computing: Soft Computing Constituents, From Conventional AI to Computational Intelligence: Machine Learning Basics.

UNIT 2

FUZZY LOGIC: Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations, Membership Functions: Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzy Expert Systems, Fuzzy Decision Making.

UNIT 3

NEURAL NETWORKS: Machine Learning Using Neural Network, Adaptive Networks, Feed forward Networks, Supervised Learning Neural Networks, Radial Basis Function Networks : Reinforcement Learning, Unsupervised Learning Neural Networks, Adaptive Resonance architectures, Advances in Neural networks

UNIT 4

GENETIC ALGORITHMS: Goals of optimization, comparison with traditional methods, schemata, Terminology in GA – strings, structure, parameter string, data structures, operators, coding fitness function, algorithm, applications of GA in Machine Learning : Machine Learning Approach to Knowledge Acquisition.

UNIT 5

Matlab/Python Lib: Introduction to Matlab/Python, Arrays and array operations, Functions and Files, Study of neural network toolbox and fuzzy logic toolbox, Simple implementation of Artificial Neural Network and Fuzzy Logic

TEXT BOOKS :

1. Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications, S.Rajasekaran, G. A. Vijayalakshami, PHI.
2. Genetic Algorithms: Search and Optimization, E. Goldberg.
3. Neuro-Fuzzy Systems, Chin Teng Lin, C. S. George Lee, PHI.
   Build_Neural_Network_With_MS_Excel_sample by Joe choong.

REFERENCE BOOKS:

1. Jyh Shing Roger Jang, Chuen Tsai Sun, Eiji Mizutani, Neuro-Fuzzy and Soft Computing, Prentice Hall of India, 2003.
2. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.
3. MATLAB Toolkit Manual
4. Timothy J.Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill
5. Goldberg, D. E, Genetic algorithm in search, optimization and machine learning, AddisonWesley, Reading Mass
6. S.N.Sivanandam, S.N.Deepa , Principles of Soft Computing, 2e, Wiley India Pvt. Ltd

MCA-215	Introduction to Machine Learning for Data Science	L T P	Cr
		3 0 0	3

OBJECTIVES

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

PRE-REQUISITES

While there are no requirements, a background in programming is strongly recommended.

COURSE OUTCOMES

Students after undergoing this course will be able to:

CO1: Identify various machine learning algorithms and terminologies and perform data pre-processing using standard ML library.

CO2: Design a pre1dictive model using appropriate supervised learning algorithms to solve any given problem.

CO3: Develop an application using appropriate unsupervised learning algorithms for performing clustering and dimensionality reduction.

CO4: Solve complex problems using artificial neural networks and kernel machines.

CO5: Implement probabilistic graphical models for suitable applications.

Unit – I:

Introduction to Machine learning for Data Science – Evolution of Data Science – Data Science Roles – Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues.

Unit – II:

Data Collection and Data Pre-Processing Data Collection Strategies – Data Pre-Processing Overview – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization.

Unit – III:

Exploratory Data Analytics Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis – Box Plots – Pivot Table – Heat Map – Correlation Statistics – ANOVA.

Unit – IV:

Model Development Simple and Multiple Regression – Model Evaluation using Visualization – Residual Plot – Distribution Plot – Polynomial Regression and Pipelines – Measures for In-sample Evaluation – Prediction and Decision Making.

Unit – V:

Model Evaluation Generalization Error – Out-of-Sample Evaluation Metrics – Cross Validation – Overfitting – Under Fitting and Model Selection – Prediction by using Ridge Regression – Testing Multiple Parameters by using Grid Search.

Text Book:

1. Essential Math for Data Science: Calculus, Statistics, Probability Theory, and Linear Algebra, by Hadrien Jean
2. A Common-Sense Guide to Data Structures and Algorithms: Level Up Your Core Programming Skills (2nd Edition), by Jay Wengrow.
3. Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python (2nd Edition), by Peter Bruce, Andrew Bruce, and Peter Gedeck.
4. Data Science for Beginners, by Andrew Park.

REFERENCES:

1. Jojo Moolayil, “Smarter Decisions : The Intersection of IoT and Data Science”, PACKT, 2016.
2. Cathy O’Neil and Rachel Schutt , “Doing Data Science”, O'Reilly, 2015.
3. David Dietrich, Barry Heller, Beibei Yang, “Data Science and Big data Analytics”, EMC 2013
4. Raj, Pethuru, “Handbook of Research on Cloud Infrastructures for Big Data Analytics”, IGI Global.

MCA-263	SOFT COMPUTING LAB	L-T-P	Cr
		0-0-2	1

EASY

1. Create a perceptron with appropriate no. of inputs and outputs. Train it using fixed increment learning algorithm until no change in weights is required. Output the final weights.
2. Create a simple ADALINE network with appropriate no. of input and output nodes. Train it using delta learning rule until no change in weights is required. Output the final weights.
3. Train the autocorrelator by given patterns: A1=(-1,1,-1,1), A2=(1,1,1,-1), A3=(-1, -1, - 1, 1). Test it using patterns: Ax=(-1,1,-1,1), Ay=(1,1,1,1), Az=(-1,-1,-1,-1).
4. Train the hetrocorrelator using multiple training encoding strategy for given patterns: A1=(000111001) B1=(010000111), A2=(111001110) B2=(100000001), A3=(110110101) B3(101001010). Test it using pattern A2.
5. Implement Union, Intersection, Complement and Difference operations on fuzzy sets. Also create fuzzy relation by Cartesian product of any two fuzzy sets and perform maxmin composition on any two fuzzy relations.

MODERATE

6. Solve Greg Viot’s fuzzy cruise controller using MATLAB Fuzzy logic toolbox.
7. Solve Air Conditioner Controller using MATLAB Fuzzy logic toolbox
8. Implement TSP using GA.
9. Write a program to implement artificial neural network without back propagation.
10. Write a program to implement artificial neural network with back propagation.

HARD

11. Implement linear regression and multi-regression for a set of data points.
12. Implement crisp partitions for real-life iris dataset.
13. Write a program to implement Hebb’s rule Write a program to implement Delta rule.
14. Write a program to implement logic gates. Week-10 CLASSIFICATION
15. Implement SVM classification by Fuzzy concepts.

MCA-255	PYTHON PROGRAMMING LAB	L T P	Cr
		0 0 2	1

COURSE OUTCOMES

CO1: Define basic concepts of python programming if statement, loops.
CO2: Define and demonstrate the use of built-in data structures "lists" and "dictionary".
CO3: Design and implement a program to solve a real-world problem
CO4: Design and implement GUI application and how to handle exceptions and files.

LIST OF PROGRAMS

Program 1: Programs using if else structure

a) Find the Largest Among Three Numbers
b) Python Program to Check Leap Year
c) Python Program to Take in the Marks of 5 Subjects and Display the Grade

Program 2: Programs using for and while loop

a) Python Program to check whether given number is Prime Number or not.
b) Python Program to Find the Sum of Digits in a Number.
c) Python Program to convert binary number to decimal number.

Program 3: Program using List and String data structure

a) Write Python Program to input a list of integers, (1) display the no of elements in the list (2) display minimum and maximum element in the list (3) display sum of square of all the element in the list (4) (5) add a new element at end and display the list (6) add a new element at given index and display list (7) display the occurrence of given element in the list (8) remove the given element in the list (9) add element from a new list from given list (10) sort the given list & reverse the given list (11) also perform slicing, concatenation and multiplication operation.
b) A fruit seller sells different type of fruits. Type of fruits and corresponding rates are stored in two different lists. Customer can order any type of fruit (one or more type) in any quantity. If total bill of customer is greater than 500, customer is given 10% discount. If any of the fruits required by the customer is not available in the store, then consider the bill amount to be -1. Write a Python program to calculate and display the bill amount.
c) Accept two strings 'string1' and 'string2' as input from the user. Generate a resultant string-1, such that it is a concatenated string of all upper case alphabets from both the strings in the order they appear. Generate a resultant string-2 that contain character which are in both string1 and 2. Print the actual resultant string-1 and resultant string-2.

Program 4: Programs using concept of list, tuple & dictionary

a) Write a Python program that take a string as input and store the character and occurrence of each character in a dictionary. Create two lists from dictionary first having each character in sorted order of their frequency and second having corresponding frequency.
b) A furniture seller sells different type of furniture, Type of Furniture and rates are stored in a dictionary. Customer can order any type of furniture (one or more type) in any quantity. If total bill of customer is greater than 10,000, customer is given 5% discount. 8% GST is charged on total bill. If any of the furniture required by the customer is not available in the store, then consider the bill amount to be -1. Write a Python program to calculate and display the bill amount.
c) Students name and their corresponding marks are stored in a dictionary. Write a Python program to perform following (1) Display name and marks of each student (2) Display the names of top two scorer (3) display the class average for this course (4) check if the

marks for given student is stored in dictionary or not, if not add the name and marks in the dictionary else display his/her marks (5) delete the name and marks of a given student in the dictionary (6) add name and marks from another dictionary and display combined dictionary.

Program 5: Program using Function in Python:

a) Write Python functions using the concept of Keyword & default arguments and write a program to use them.
b) Write python functions to use the concept of variable length argument & global variable.

Program 6: Program using concept of Class, object, class variable, class method:

a) Create a class Account with name, account no and balance as attribute and no_of_accounts as class variable. Account no should be generated automatically (starting from 1) using the class variable no_of_account. Add the methods for displaying the account information, depositing given amount, withdrawing given amount and initializer method to initialize the object. Create objects of Account class and call different method to test the class.
b) Create a class Employee with name , empid ,salary as attribute and no_of_ employee and annual_incr (% annual increment) as class variable. empid should be generated automatically (starting from 1) using the class variable, no_of_employee. Add the instance methods for displaying the employee information, annually increasing the salary with help of class variable annual_incr , class method to change the value of annual_incr and initializer method to initialize the object. Create objects of employee class and call different method to test the class (program using class method).

Program 7: Program using the concept of Inheritance

a) Create a class Polygon to represent a polygon having no of sides and a list having magnitude of each side as attribute. Add the inputSides() to input sides and displaySides() to display sides as methods. Derive a class Triangle from Polygon and add an additional method displayArea() to display area. Create object of Triangle and call different methods to test the class.

b) Create a class Person having name, age, as attributes, __init__() method to initialize the object and display() to display person information. Derive a class Student from Person having roll no, University name, branch as additional attributes and __init__(), display() to display student information and change_Branch() method. Create object of Student type and call different methods to test the class.

Program 8: Program using the concept of Polymorphism, Operator Overloading

a) In a retail outlet there are two modes of bill Payment (1) Cash : Calculation includes VAT(10%) Total Amount = Purchase amount + VAT (2) Credit card: Calculation includes processing charge and VAT Total Amount = Purchase amount + VAT (10%) + Processing charge (2%) The act of bill payment is same but the formula used for calculation of total amount differs as per the mode of payment. Can the Payment maker simply call a method and that method dynamically selects the formula for the total amount? Demonstrate this Polymorphic behaviour with code.

b) Write a program to create a class to represent length in feet and inch. Overload the “+” operator to add the two object of length type.

Program 9: Program on file handling and Exception handling in Python

a) Write a python program to write few lines on a file, read it back and create a dictionary having each word in file as keys in dictionary and occurrence of these word as values and print the dictionary.

b) Write a function divide (arg1, arg2) to divide arg1 by arg2. Use the exception handling mechanism to handle all type of possible exceptions that may occur. Take the value of arg1 and arg2(of any type) from user as input and call the function divide to print the result of division or suitable message if any type of exception occurs(use also else and finally block).

Program 10: Program on Data Analysis and Visualization

a) Importing and Exporting Data, Basic Insights from Datasets, Cleaning and Preparing the Data: Identify and Handle Missing Values.

b) Basic Visualization Tools, Specialized Visualization Tools, Creating and Plotting Maps.

MCA-265	Introduction to Machine Learning for Data Science Lab	L T P	Cr
		0 0 2	1

OBJECTIVES

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

PRE-REQUISITES

While there are no requirements, a background in programming is strongly recommended.

Lab experiments:

1. Installation of Python Libraries/ MATLAB tools for Machine Learning.
2. Data pre-processing using Python Machine Learning libraries/ MATLAB.
3. Design a model to predict the housing price from Boston Dataset using Multivariate Linear Regression.
4. Build a classifier using Logistic Regression, k- Nearest Neighbour / Decision Tree to classify whether the given user will purchase a product or not from a social networking dataset.
5. Segment a customer dataset based on the buying behaviour of customers using K-means/Hierarchical clustering.
6. Dimensionality reduction of any CSV/image dataset using Principal Component Analysis.
7. Recognition of MNIST handwritten digits using Artificial Neural Network.
8. Build an email spam classifier using SVM.
9. Classify the given text segment as ‘Positive’ or ‘Negative’ statement using the Naïve Bayes Classifier.
10. Predict future stock price of a company using Monte Carlo Simulation.
11. Demonstrate performing classification on data sets Classification Tab.
12. Load each dataset into Weka and run id3, j48 classification algorithm, study the classifier output. Compute entropy values, Kappa statistic.
13. Compare classification results of ID3,J48, Naïve-Bayes and k-NN classifiers for each dataset , and reduce which classifier is performing best and poor for each dataset and justify.
14. Demonstrate performing clustering on data sets Clustering Tab.
15. Explore other clustering techniques available in Weka.
16. Prediction and Analysis of Student Performance by Data Mining in WEKA.

Elective I

MCA-122A	ADVANCE DATABASE MANAGEMENT SYSTEM	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To provide knowledge about various organizations and management information systems, keeping in view the aspects of shareability, availability, evolvability and integrity.

COURSE OUTCOMES

The students undergoing this course will be able:
CO1: Recap to use DBMS features and be familiar with advanced SQL usage
CO2: Understanding of Query Processing and Query Optimization
CO3: Be proficient with Transactions, Concurrency Control and Recovery Systems
CO4: Be exposed to parallel, distributed and deductive databases and object database systems
CO5: To learn about the Database System Architectures.

UNIT I

Coping with System Failures: Introduction to ADBMS, ACID properties, Issues and Models for Resilient Operation, Undo Logging, Redo Logging, Undo/Redo Logging, Logging Rules, Recovery using different Logging methods, Quiescent and Non-quiescent Check pointing a Log, Recovery with a checkpointed Log, Protecting against Media Failures, Non-quiescent Archiving, Recovery using an Archive and Log, Transactions in SQL Serializability, Atomicity, Read-only Transactions, Dirty Reads, other Isolations Levels, Review of PL/SQL

UNIT II

Concurrency Control: Serial and Serializable Schedules, Conflict-Serializability, Precedence Graphs and a Test for Conflict-Serializability, Enforcing Serializability by Locks, The Locking Scheduler, Two-Phase Locking (2PL), Locking Systems with several Lock Modes: shared and Exclusive Locks, Compatibility Matrices, Upgrading Locks, Update Locks, Increment Locks, An architecture for a Locking Scheduler, The Lock Table, Managing Hierarchies of Database Elements: Locks with Multiple Granularity, The Tree Protocol, Concurrency Control by Timestamps, Concurrency Control by Validation, Constraints and Triggers.

UNIT III

Advanced Transaction Management: Serializability and Recoverability, Recoverable Schedules, ACR, Logical Logging, Recovery from Logical Logs, View Serializability, Polygraphs and the Test for View-Serializability, Resolving Deadlocks, Deadlock Prevention by Ordering Elements and Timestamps, Distributed Databases: Distributed Commit, Two-phase Commit (2PC), Distributed Locking, Long-duration Transactions, Sagas and Compensating Transactions

UNIT IV

The Query Compiler: Parsing, Estimating the cost of operations, Query optimization, Completing the Physical-Query-Plan and Query Execution; Storage management.

UNIT V

Database System Architectures: Object Definition Language (ODL), Object-relational Model, XML and its Data Model, Object-orientation in Query Languages, Logical Query Languages, Centralized and Client-Server Architectures, Parallel Databases, Spatial and Geographic Databases, Multimedia Databases, Mobility and Personal Databases.

TEXT BOOK

1. Silberschatz A., Korth H. F. and Sudarshan S., “Database System Concepts”, 3rd edition, McGraw-Hill, International Edition, 1997.
2. S. Chand, Rajiv Chopra, “DBMS(A practical approach)”, Revised edition, Pearson.
3. H. Garcia-Molina, J. D. Ullman, and J. Widom: Database Systems: The Complete Book. Pearson

REFERENCE BOOKS

1. R. Ramakrishnan & J. Gehrke, Database Management Systems. Mc Graw Hill
2. Date C. J., “An Introduction to Database Systems”, 7th edition, Addison- Wesley, Low Priced Edition, 2000.
3. Desai Bipin, “Introduction to Database Management System”, Galgotia Publications, 1991.
4. Elmasri R. and Navathe S. B., “Fundamentals of Database Systems”, 3rd edition, Addison-Wesley, Low Priced Edition, 2000.

MCA-122B	BLOCKCHAIN TECHNOLOGY	L	T	P	Credit
		3	0	0	3

OBJECTIVE:

- To provide conceptual understanding of the function of Blockchain as a method of securing distributed ledgers.
- To understand the structure of a blockchain and why/when it is better than a simple distributed database
- To make students understand the technological underpinnings of blockchain operations as distributed data structures and decision making systems.
- To understand a “smart” contract and its legal implications.
- To provide a critical evaluation of existing “smart contract” capabilities and platforms, and examine their future directions, opportunities, risks and challenges.

COURSE OBJECTIVES:

After completion of course, students would be able to:

1. **CO1:** Able to design smart contracts and decentralized applications.
2. **CO2:** Able to understand Distributed Ledger Technologies and apply this concept in real world.
3. **CO3:** Able to design innovative application models using the Blockchain technology.
4. **CO4:** Able to understand core concepts of Blockchain technology that are commonly used across multiple industries to solve large scale problems.

Unit 1: Introduction

Introduction to Blockchain, Structure of a Block, Types of Blockchain, Public Ledgers, Blockchain as public ledgers, Crypto currency as application of blockchain technology, **Introduction to Bitcoin**, History of Bitcoin, Bitcoin Transactions, Bitcoin Mining, Bitcoin Address.

Introduction to Ethereum - Ethereum Virtual Machine (EVM), Wallets for Ethereum, Differences between Ethereum and Bitcoin, Block format, Mining algorithm, Solidity, Smart Contracts, Some attacks on smart contracts.

Unit 2: Basic Cryptographic Primitives used in Blockchain

Secure, Collision-resistant hash functions, Digital signature, Public key cryptosystems, Zero-knowledge proof systems Cryptographic Hash Function, SHA-256, Properties of a hash function, Hash pointer and Merkle tree.

Unit 3: Consensus Algorithms

Consensus, Distributed consensus in open environments, Consensus in a Bitcoin network
Types of consensus algorithm: Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Ripple, Proof of Burn

Unit 4: Blockchain Technology

Blockchain Technology: Hyper ledger Fabric: System architecture, ledger format, chaincode execution, transaction flow and ordering, private channels, membership service providers, Fabric Peer and Certificate Authority, Case studies of applications

Unit 5: Research-based Study

The advances and the latest trends in the course as well as the latest applications of the areas covered in the course. The latest research conducted in the areas covered in the course. Discussion of latest research published in IEEE/ACM transactions, SCI/SCIE/Web of Science/SCOPUS indexed journals and Tier-1 conference of this area. Discussion on some of the latest products available in the market based on the areas covered in the course and patents filed in the areas covered.

Text-Books

1. Mastering Bitcoin: Unlocking Digital Cryptocurrencies, by Andreas Antonopoulos, O’Reilly publisher
2. Blockchain Blueprint for a New Economy,by Melanie Swan, O’Reilly
3. Narayanan, Arvind, et al. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press, 2016.
4. Antonopoulos, Andreas M. Mastering Bitcoin: Programming the Open Blockchain. O’Reilly Media, Inc., 2017
5. Antonopoulos, Andreas M. and Wood, Gavin. Mastering Ethereum. O’Reilly Media, Inc., 2018. (Free draft available at https://github.com/ethereumbook/ethereumbook)
6. Ethereum project documentation. Online: http://www.ethdocs.org/en/latest/
7. S. Shukla, M. Dhawan, S. Sharma, S. Venkatesan, ‘Blockchain Technology: Cryptocurrency and Applications’, Oxford University Press, 2019.
8. Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits - https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html
9. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

MCA-122C	Theory of Computation	L T P	Cr
		3 0 0	3

COURSE OBJECTIVE

To understand the theory and practice of compiler implementation. To learn finite state machines and lexical scanning. To learn context free grammars, compiler parsing techniques, construction of abstract syntax trees, push down Automata and Turing Machine.

PRE-REQUISITES

The students are expected to have a strong background in the fundamentals of discrete mathematics (symbolic logic, set, induction, number theory, summation, series, combinatorics, graph, recursion, basic proof techniques, etc.), algorithms and data structures.

COURSE OUTCOME:

After completing this course, the students will be able to:

CO1: Understand the relation between types of languages and types of finite automata

CO2: An ability to design grammars and automata for different language classes.

CO3: Understanding the Context free languages and grammars, and also Normalizing CFG.

CO4: Understand the concept of Pushdown automata and its application.

CO5: To understand basic properties of Turing machines and computing with Turing machines.

UNIT I: FINITE AUTOMATA AND REGULAR EXPRESSIONS

Finite state systems; basic definitions non-deterministic finite automata (NDFA), deterministic finite automata (DFA), equivalence of DFA and NDFA, limitations of FSM, Moore and Mealy Machines; Equivalence of Moore and Mealy Machines., Minimization of Finite Automata.

UNIT II: PROPERTIES OF REGULAR SETS

Regular expressions; equivalence of finite automata and regular expressions, regular expression conversion and vice versa, Arden's theorem; The Pumping Lemma for regular sets; applications of the pumping lemma; closure properties of regular sets.

UNIT III: CONTEXT FREES GRAMMARS & GREIBACH NORMAL FORM

Definition, Context free and context sensitive grammar; ambiguity regular grammar; reduced forms; Chomsky Normal Form (CNF), Greibach Normal Form (GNF).

UNIT IV: PUSHDOWN AUTOMATA & TURING MACHINES

Introduction to pushdown machines; design of PDA; conversion of PDA to CFG and vice versa, application of pushdown machines.

Basic concepts of Turing machines, Deterministic and non-deterministic Turing machines; design of Turing machines; halting problem of Turing machines.

UNIT V:

DECIDABILITY AND RECURSIVELY ENUMERABLE LANGUAGE

Definition of an Algorithm, Decidability, Decidable Languages, Undecidable Languages, Haling Problem of Turing Machine, Post Correspondence Problem.

TEXT BOOK

1. Hopcroft, Ullman O. D. and Mothwani R., “Introduction to Automata Theory, Language & Computations”, Addison Wesley, 2001
2. Mishra K. L. P. and Chandrasekaran N., “Theory of Computer Science - Automata, Languages and Computations”, Prentice Hall of India, 2000

REFERENCE BOOKS

1. Linz Peter, “Introduction to Formal Languages & Automata”, Narosa Publications, 2001
2. Greenlaw Ramond and Hoover H. James, “Fundamentals of the Theory of Computation - Principles and Practice”, Harcourt India Pvt. Ltd., 1998
3. Lewis H. R. and Papaditriou C. H., “Elements of Theory of Computation”, Prentice Hall of India, 1998
4. Martin John C., “Introduction to Languages and Theory of Computations”, Tata McGraw Hill, 2003

ELECTIVE II

MCA-217A	COMPILER DESIGN	L-T-P	Credits
		3-0-0	3

COURSE OBJECTIVE

To make the student to understand the process involved in a compiler, create an overall view of various types of translators, linkers, loaders, and phases of a compiler, understand what is syntax analysis, various types of parsers especially the top down approach, awareness among students the various types of bottom up parsers, understand the syntax analysis and, intermediate code generation, type checking, the role of symbol table and its organization, Code generation, machine independent code optimization and instruction scheduling.

PRE-REQUISITES

Knowledge of automata theory, context free languages, computer architecture, data structures and simple graph algorithms, logic or algebra.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Student will be familiar with the front-end as well as back-end stages of compiler design and Design Lexical analyzer for given language using LEX tools

CO2: understand the differences between constructing lexers/parsers by hand versus using automated generators

CO3: Hands-on experience with generating intermediate representations, which in turn will let them appreciate the importance of designing simpler languages

CO4: To appreciate the nuances of analyzing and transforming programs for performance

CO5: Experience of working with relatively large programming environments, which will also inculcate a sense of good software design

UNIT I

INTRODUCTION TO COMPILING & LEXICAL ANALYSIS: Introduction of Compiler, Major data Structure in compiler, BOOT Strapping & Porting, Compiler structure: analysis-synthesis model of compilation, various phases of a compiler, Lexical analysis: Input buffering, Specification & Recognition of Tokens, LEX

UNIT II

SYNTAX ANALYSIS: Basic Parsing Techniques: Parsers, Shift reduce parsing, Operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

UNIT III

SYNTAX DIRECTED TRANSLATION: Syntax directed definitions: Construction of Syntax trees, bottom up evaluation of S-attributed definition, L-attribute definition, Top down translation, Bottom Up evaluation of inherited attributes Recursive Evaluation, Analysis of Syntax directed definition.

UNIT IV

CODE GENERATION: Intermediate code generation: Declarations, Assignment statements, Boolean expressions, Case statements, Procedure calls Code Generation: Issues in the design of code generator, Basic block and flow graphs, register allocation and assignment, DAG representation of basic blocks, peephole optimization, generating code from DAG.

UNIT V

CODE OPTIMIZATION: Introduction to Code optimization: sources of optimization of basic blocks, loops in flow graphs, dead code elimination, loop optimization, Introduction to global data flow analysis, Code Improving transformations, Data flow analysis of structure flow graph Symbolic debugging of optimized code.

TEXTBOOK:

1. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques andTools , Pearson Education
2. Raghavan, Compiler Design, TMH Pub.

REFERENCES

3. Louden. Compiler Construction: Principles and Practice, Cengage Learning
4. Holub. Compiler Design in C, Prentice-Hall Inc., 1993.
5. Mak, writing compiler & Interpreters, Willey Pub.

MCA-217B	.NET FRAMEWORK USING C#	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To equip students with C# programming Concepts

PRE-REQUISITES

Knowledge of C programming language.

COURSE OUTCOMES

The students undergoing this course will be able:
CO1: To know the basics of Dot net programming language
CO2: To learn about the basics of C# programming
CO3: To learn about objects and classes in C# programming language
CO4: To know about the exception handling and delegates in C# programming
CO5: To learn about different database connectivity like ADO etc.

UNIT I

PHILOSOPHY OF .NET AND ITS MAJOR COMPONENTS: Origin of .NET technology; .NET platform; benefits and limitations of .NET; building blocks of .NET framework; .NET programming languages; .NET types and namespaces; Understanding CLR, CTS and CLS; developing C# Applications using Visual Studio .Net

UNIT II

UNDERSTANDING C#: Data Types, Variables & Constants, Operators in C#, Arithmetic Operators, Prefix and Postfix notation, Assignment Operators, Relational Operators, Other Operators, Operators precedence, Flow Control and Conditional Statements if-else statement, switch statement, Loops in C#, for loop, do-while loop, Array in C#, foreach Loop, Comparison among C++; Java and C#; benefits of C#; object-oriented programming using C#

UNIT III

OBJECTS AND CLASSES: Concept of a class, Objects, Fields, Methods, Instantiating the class, Accessing the members of a class, Access modifiers, Properties, Static members of the class, Constructors, Desturctors, Implementing inheritance in C#, The base keyword, Protected Access Modifier, sealed keyword, Polymorphism, using the reference of the base type for referencing the objects of the child class, Overriding the methods, the new keywords, Type casting, is and as keywords.

UNIT IV

EXCEPTION HANDLING AND DELEGATES: Exceptions in C# and .Net, Handling Exceptions using the try-catch-finally blocks, Delegates Basics, Delegates in the .Net Framework, Passing delegates to methods, Multicast Delegates.

UNIT V

ADO .NET AND ASP.NET: Comparison of ADO and ADO. NET, Introduction to data access with ADO.NET, Components of ADO.NET; Comparison of ASP and ASP .NET, Features of ASP .NET, Features provided by ASP .NET; web forms and their components.

TEXT BOOKS

1. Balaguruswammy, E, "Programming in C#", Tata McGraw Hill
2. Gunnerson Eric, "A programmer's Introduction to C#", IDG Books

REFERENCE BOOKS

1. Jain, V K, "The Complete Guide to C# Programming", IDG Books India.
2. Pappas & Murray, "C# Essentials", Prentice Hall of India
3. Wakefield, "C# and .NET Web Developers Guide", IDG Books India.

MCA-217C	Mobile Communication and android application development	L T P	Cr
		3 0 0	3

OBJECTIVES

Android is the most popular Mobile OS, having the largest installed base and is a market leader in Mobile Technology. There is a huge demand for skilled Android Developers all over the world. Most businesses across multiple domains are building Android Apps both for enterprise and retail products. Whether you are a student or an IT Professional, possessing Android Development skills will help you take the next big leap in your career. The Android Development course is primarily designed for beginners and experienced programmers who want to learn how to create applications in Android. The step-by-step lessons at Geosys start from Java Essentials for Android and cover all that you need to develop professional Android Apps. There are chapter wise quizzes and coding assignments after each unit to help and reinforce your understanding.

PRE-REQUISITES

While there are requirements of java programming, a background in programming is strongly recommended.

COURSE OUTCOMES

Students after undergoing this course will be able to:

CO1: Understanding of Basic Android Development tools such as Eclipse, DDMS, Drawable, Listeners, and so on.
CO2: How to use various Layouts and Widgets in Android Applications.
CO3: Using the protocols to create Communication and Media in android Systemes.
CO4: Understanding the Storage Techniques and Animation in Android.
CO5: Use development tools, such as those found in the Android Developer's Toolkit to efficiently create, understand, debug and optimize Android applications. Understand the key forces and constraints acting on handheld devices and know how to accommodate these when designing and building their own Android applications.
CO6: Understand the Android platform's organization, patterns and programming mechanisms and be able to use them effectively to develop their own Android applications.

Unit I :

Basics of Android Learning Objectives - By the end of this week you will have a basic understanding of Android development tools i.e. - Eclipse, DDMS etc. and you will have a clear vision about what Android manifest file does and the importance of the activity lifecycle. Topics - Eclipse, DDMS, Activity LifeCycle, Manifest File, Locales, Drawables, Listeners, Supporting Multiple Screens.

Unit II:

Android Layouts and Widgets Learning Objectives – After this week you will be able to use various layouts and widgets in your Android Applications. Furthermore, you can create list and grid views in your program and populate them with data sources. Topics - Linear Layout, Relative Layout, Table Layout, Grid View, List View, Toggle Buttons, Checkboxes, Radio Buttons, Radio Group, Spinner, Autocomplete TextView, Web Views

Unt III:

Communication and Media Learning Objectives - In this week you will be able to create applications with multiple activities and you can pass information between multiple activities. Also you will be able to use audio and video files in your project and generate notifications in android. Topics - Date Picker, Time Picker, Intents, Bundle, Switching between Activities, Shared Preferences, Notifications in Android, Media Player, Video Player

Unit IV:

Storage Techniques and Animation in Android Learning Objectives - By the end of this week you will be able to create applications which can make use of internal and external storage (SD Card). You will also learn how to use Animation in android and an interesting DrawingBrush Application will be explained in its very detail. Topics - Internal Storage, External Storage, Frame animation, Tween animation, Canvas, Paint and Path

Unit V:

Web Services and Customizations of Widgets Learning Objectives -By the end of this week you will be able to parse JSON data in your applications. Also in this class instructor will teach you the customization of different widgets that includes buttons, spinners and most importantly list and grid views. Topics - HTTPClient, HTTPResponse, JSON, Tweeter App with JSON, Async Task, Customizing a button, Customizing Spinners, Customizing ListView, Customizing GridView

Unit VI:

Advanced Android Concepts Learning Objectives - You will be able to create android applications that can make use of SQLite database by the end of this week. Also you will publish your First App on Google Play. Topics - Starting Activity For Result, SQLite Database, SQLite Data Types, SQLiteOpenHelper Class, Content Values, Cursor, Content Providers, Publish your App on Google Play.

TextBook:

1. Headfirst Android Development Dawn Griffiths 1st edition O'Reilly
2. Android Programming for Beginners John Horton 2nd edition Packt Publishing
3. Android Programming with Kotlin for Beginners John Horton 1st edition Packt Publishing Limited
4. Head -First Kotlin Dawn Griffiths 1st edition O'Reilly Media
5. Android App Development FDMichael Burton 3rd edition For Dummies