

**DR. RAMMANOHAR LOHIA AVADH UNIVERSITY
AYODHYA U.P.**



**Evaluation Scheme & Syllabus for
Master of Computer Application
of First Year
On
Choice Based Credit System**

(Effective from the Session: 2020-21)

Dr. RamManohar Lohia Avadh University, Ayodhya U.P.

Study and Evaluation Scheme

MCA (Master of Computer Applications)

(Effective From Session 2020-21)

Year – I Semester – I

Sl. No.	Subject Code	Subject Name	Periods			Evaluation Scheme					Credit
			L	T	P	Sessional Exams			ESE	Subject Total	
						CT	TA	Total			
	THEORY SUBJECT										
1	MCA 101	Programming in ‘C’ & Data Structure	3	1	0	30	20	50	100	150	04
2	MCA 102	Computer Organization & Architecture	3	1	0	30	20	50	100	150	04
3	MCA 103	Object oriented system with C++	3	1	0	30	20	50	100	150	04
4	MCA 104	Operating Systems	3	1	0	30	20	50	100	150	04
5	MCA 105	Programming with MATLAB	3	1	0	30	20	50	100	150	04
6	MCA 106	Cyber Security & Information System	3	1	0	30	20	50	100	150	04
	Practical										
7	MCA 107	Programming in ‘C’ & Data Structure Lab	0	0	3	10	10	20	30	50	02
8	MCA 108	Computer Organization Lab	0	0	3	10	10	20	30	50	02
9	MCA 109	Object oriented system with C++ Lab	0	0	3	10	10	20	30	50	02
10	MCA 110	Operating Systems Lab	0	0	3	10	10	20	30	50	02
11	MCA 111	Programming with MATLAB Lab	0	0	3	10	10	20	30	50	02
12	MCA 112	Project / Seminar*	0	0	3	10	10	20	30	50	02
		Total	18	6	18	-	-	-	-	1200	36

CT: Class Test

TA: Teacher Assessment

L/T/P: Lecture/Tutorial/ Practical

MSE: Mid Semester Examination

ESE: End Semester Examination

*Seminar means giving explanation for a group of people. Paper presentation means presenting an explanation on a specified concept may include seminar in it.

*Project means doing for result oriented it may include both above two.

Students make a project thesis in specific topic which you read in this Semester. Maximum Two Students may be participated in one Project. After developing Project thesis student presenting an explanation on a specified concept may include seminar in it. Students develop project under the guidance of faculty members.

Prepare Project Thesis according Appendix 1

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER I	MCA 101	PROGRAMMING IN 'C' & DATA STRUCTURE	L	T	P	C
			3	1	3	4
GOAL	1. To introduce programming in C and also explore the power of computational techniques that is currently used by engineers and scientists and to develop programming skills with reasonable complexity. 2. To provide an in-depth knowledge in problem solving techniques using data structures.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Gain knowledge in C programming. 2. Learn calculating the efficiency of algorithms. 3. Understand the concepts and applications of stack and queue data structures. 4. Learn various sorting and searching techniques. 5. Understand the concepts and applications of tree and graph data structures. 6. Have a good understanding of problem solving using data structure.			The student should be able to 1. Develop applications using C programming 2. Calculate the time complexity of algorithms 3. Implement stacks and queues for various applications. 4. Implement tree data structure for different applications. 5. Implement various sorting and searching techniques. 6. Apply the concepts of graph for computing shortest path and construct MST.			

UNIT I:

Introduction to C: History and Structure of C Programming, Standard I/O in "C", C Declaration.

Fundamental Data Types and Storage Classes: Character types, Integer, short, long, unsigned, storage classes, automatic, register, static and external,

Operators and Expressions: Using numeric and relational operators, mixed operands and type conversion, Logical operators, Bit operations, Operator precedence and associability

Conditional Program Execution: Applying if and switch statements, nesting if and else, restrictions on switch values, use of break and default with switch

Program Loops and Iteration: Uses of while, do and for loops, multiple loop variables, assignment operators, using break and continue

Modular Programming: Passing arguments by value, scope rules and global variables, separate compilation, and linkage, building your own modules.

UNIT II:

Arrays: Array notation and representation, manipulating array elements, using multidimensional arrays, arrays of unknown or varying size, Array Insertion, Deletion and Traversing Sequential search, Address calculation, Strings

Structures: Purpose and usage of structures, declaring structures, assigning of structures.

Union: Purpose and usage of union, declaring union, assigning of union,

Pointers to Objects: Pointer and address arithmetic, pointer operations and declarations, using pointers as function arguments, Dynamic memory allocation

The Standard C Preprocessor: Defining and calling macros, utilizing conditional compilation, passing values to the compiler

The Standard C Library: Input/Output : fopen, fread, etc, string handling functions, Math functions : log, sin, alike Other Standard C functions.

UNIT III:

Data Structure Introduction: Basic Terminology, Elementary Data Organization, Data Structure operations, Algorithm Complexity and Time-Space trade-off

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks,

Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack.

Recursion: Recursive definition and processes, recursion in C, example of recursion, Tower of Hanoi Problem, simulating recursion. Backtracking, recursive algorithms, principles of recursion, tail recursion, removal of recursion.

Queues: Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty. Circular queue, Dequeue, and Priority Queue.

Linked list: Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list, Garbage Collection and Compaction.

UNIT IV:

Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree. Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees. Traversing Threaded Binary trees, Huffman algorithm.

Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-trees.

Graphs: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

UNIT V:

Searching and Hashing: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.

Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting.

File Structures: Physical Storage Media File Organization, Organization of records into Blocks, Sequential Files, Indexing and Hashing, Primary indices, Secondary indices, B+ Tree index Files, B Tree index Files, Indexing and Hashing Comparisons.

Text Books :

1. Gottfried, "Programming in C", Schaum's Series, Tata McGraw Hill
2. Kernighan, Ritchie, "The C Programming Language", PHI
3. Yashwant Kanitkar, "Working with C", BPB
4. Y. Langsam, M. J. Augenstein and A. M. Tanenbaum, —Data Structures using C++, Pearson Education Asia, 2004
5. Yashwant Kanitkar, "Let us C", BPB
6. Bajpai, Kushwaha, Yadav, "Computers & C Programming", New Age
7. E. Balagurusamy, "Programming in ANSI C", TMH
8. Ellis Horowitz, S. Sahni, Freed, —Fundamentals of Data Structures in C, 2nd edition, 2007
9. Mark Allen Weiss, —Data Structures and Algorithm Analysis in C++, Pearson, 1996, 2nd edition.
10. S. Lipschutz, —Data Structures, McGraw Hill, 1986.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER II	MCA 102	COMPUTER ORGANIZATION & ARCHITECTURE	L	T	P	C
			3	1	3	4
GOAL	To learn the basic methods and provide the fundamental concepts used in the design of digital systems.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. Acquire the knowledge about computer hardware and software. 2. Learn number systems, codes, basic postulates of Boolean algebra and shows the correlation between Boolean expressions. 3. Gain knowledge of the methods for simplifying Boolean expressions. 4. Outline the formal procedures for the analysis and design of combinational circuits, 5. Learn about several structural and behavioral models for synchronous sequential circuits, 6. Provide knowledge of the concept of memories and programmable logic devices. 7. Learn the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division. 8. Study in detail the different types of control and the concept of pipelining. 9. Study the hierarchical memory system including cache memories and virtual memory. 10. Study the different ways of communicating with I/O devices and standard I/O interfaces.			At the end of the course the student should be able to: 1. Understand the functions of digital computer. 2. Reduce complex logical expressions using various postulates of Boolean algebra. 3. Use different graphical methods for the simplification of complex logical expressions. 4. Use the design methodology for combinational logic circuits. 5. Make use of design concepts of sequential circuits. 6. Understand the structure of various semiconductor storage devices. 7. Illustrate the basic arithmetic and logic operations in the computer. 8. Explain memory organisation. 9. Describe I/O interfacing 10. Explain device subsystems			

UNIT I:

Introduction to Computer Fundamentals: Evolution of Computers, Generations of Computer Classification of Computers, Application of Computers, Components of a Computer System, Hardware, Software, Starting a Computer (Bootng)

Data Representation in Computer Systems: Introduction, Positional Numbering Systems, Converting Between Bases, Signed Integer Representation, IEEE Floating-Point Representation, Character Codes

Arithmetic: Overview, Fixed Point Addition and Subtraction, Fixed Point Multiplication and Division, Floating Point Arithmetic

Boolean Algebra and Digital Logic: Introduction, Boolean Algebra, Boolean Expressions, Boolean Identities, K-Maps & Map minimization , Logic Gates, Digital Components, Combinational Circuits, Sequential Circuits

UNIT II:

Memory, Register and Register transfer : Register, Register Transfer Language, Bus and Memory Transfers, Bus Architecture, Bus Arbitration, Arithmetic Logic, Shift Micro-operation, Arithmetic Logic Shift Unit, Design of Fast address, Arithmetic Algorithms (addition, subtraction, Booth Multiplication)

Memory Hierarchy, Main Memory (RAM and ROM Chips), Auxiliary memory, Cache memory, Virtual Memory, Memory management hardware

UNIT III:

Control Design: Hardwired & Micro Programmed (Control Unit): Fundamental Concepts (Register Transfers, Performing of arithmetic or logical operations, Fetching a word from memory, storing a word in memory), Execution of a complete instruction, Multiple-Bus organization, Hardwired Control, Micro programmed control(Microinstruction, Microprogram sequencing, Wide-Branch addressing, Microinstruction with Next-address field, Prefetching Microinstruction).

UNIT IV:

Processor Design: Processor Organization: General register organization, Stack organization, Addressing mode, Instruction format, Data transfer & manipulations, Program Control, Reduced Instruction Set Computer.

Input-Output Organization:I/O Interface, Modes of transfer, Interrupts & Interrupt handling, Programmed I/O, Direct Memory access, Input-Output processor, Serial Communication.

UNIT V:

Device Subsystems: External storage systems; organization and structure of disk drives and optical memory; Basic I/O controllers such as a keyboard and a mouse; RAID architectures; Video control; I/O Performance; SMART technology and fault detection; Processor to network interfaces

RISC & CICS Architecture, Basic MIPS Implementation, Pipelining, Instruction-level Parallelism, Parallel Processing Challenges, Flynn's Classification, Hardware Multi-threading, Multicore processing.

Text-Books(TB)

1. Logic and Digital Design, **Morris mano and Kimicharels 4th Edition, Prentice Hall.**
- 2.Computer System Architecture, M. Mano(PHI)
- 3.Computer Organization, Vravice, Zaky&Hamacher (TMH Publication)

Reference Books (RB)

1. Structured Computer Organization, Tannenbaum(PHI)
2. Computer Organization, Stallings(PHI)
3. Computer Organization, John P.Hayes (McGraw Hill)

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER III	MCA 103	OBJECT ORIENTED SYSTEM WITH C++	L	T	P	C
			3	1	3	4
GOAL	1. To learn the methodologies of object oriented analysis and design. 2. This course advances students' knowledge in problem solving and programming principles for scientific and technical applications through the presentation of object-oriented programming in the C++ language. The course emphasizes data abstraction and object oriented programming design through the implementation, in C++, of classes and numerous related concepts. This includes inheritance, polymorphism, and inter-object communication, as well as techniques with which you can generalize classes, such as templates and operator overloading.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. Learn the Object oriented life cycle. 2. Know how to identify objects, relationships, Services and attributes through UML. 3. Understand the Use case diagrams. 4. Understand object oriented analysis and design 5. Know and apply the object oriented design process, software quality and usability. 6. Learn Object-oriented programming paradigm. 7. Understand advanced object-oriented features through C++ programming language. 8. Learn exception handling. 9. Learn Generic classes and templates.			At the end of the course the student should be able to: 1. Acquire knowledge of OOAD. 2. Demonstrate the design concepts using UML diagrams. 3. Practice through object oriented life cycle. 4. Draw UML diagrams 5. Able to design application using OOAD tools. 6. Explain concepts in object oriented programming. 7. Write simple programs in C++. 8. Demonstrate the concept of functions, operator overloading, inheritance through C++ programs. 9. Demonstrate the concepts of exception handling, generic functions, and templates.			

UNIT I:

Introduction: An Overview of Object Oriented Systems Development, Object Basics, Object Oriented Systems Development Life Cycle.

Object Oriented Methodologies: Rumbaugh Methodology, Booch Methodology, Jacobson Methodology, Patterns, Frameworks, Unified Approach, Unified Modeling Language, Use case, class diagram, Interactive Diagram, Package Diagram, Collaboration Diagram, State Diagram, Activity Diagram.

Object Oriented Analysis: Identifying use cases, Object Analysis, Classification, Identifying Object relationships, Attributes and Methods.

Object Oriented Design: Design axioms, Designing Classes, Access Layer, Object Storage, Object Interoperability.

UNIT II:

Object Modeling: Object & classes, Links and Associations, Generalization and Inheritance, Aggregation, Abstract classes, A sample object model, Multiple Inheritance, Meta data, candidate keys, constraints.

Dynamic Modeling: Events and States, Operations and Methods, Nested state Diagrams, Concurrency, Relation of Object and Dynamic Models, advanced dynamic model concepts, a sample dynamic model.

UNIT III:

Functional Modeling: Functional Models, Data flow Diagrams, Specifying Operations, Constraints, a sample functional model.

Translating object oriented design into an implementation, OMT Methodologies, examples and case studies to demonstrate methodology, comparison of Methodology, SA/SD, and JSD, Application design and development using OOSD tools.

UNIT IV:

Programming in C++ : Introduction OOP Paradigm: Comparison of Programming paradigms, Characteristics of Object-Oriented Programming Languages, Brief History of C++, Structure of a C++ program, Difference between C and C++ - cin, cout, new, delete operators, ANSI/ISO Standard C++, Comments, Working with Variables and const Qualifiers. Enumeration, Arrays and Pointer, Default Parameter Value, Using Reference variables with Functions.

Functions and Overloading: Abstract data types, Class Component, Object & Class, Constructors Default and Copy Constructor, Assignment operator deep and shallow coping, Access modifiers – private, public and protected. Implementing Class Functions within Class declaration or outside the Class declaration. Instantiation of objects, Scope resolution operator, working with Friend Functions, Using Static Class members. Understanding Compile Time Polymorphism function overloading Rules of Operator Overloading (Unary and Binary) as member function/friend function, Implementation of operator overloading of Arithmetic Operators, Overloading Output/Input, Prefix/ Postfix Increment and decrement Operators, Overloading comparison operators, Assignment, subscript and function call Operator , concepts of namespaces.

UNIT V:

Inheritance and Polymorphism: Inheritance, Types of Inheritance, Abstract Classes, Overriding inheritance methods, Constructors and Destructor in derived classes. Multiple Inheritance.

Polymorphism: Polymorphism, Type of Polymorphism – compile time and runtime, Understanding Dynamic polymorphism: Pointer to objects, Virtual Functions (concept of VTABLE) , pure virtual functions, Abstract Class.

Exception Handling and Files: Understanding of working and implementation of Exception Handling, Advanced Input/Output, Manipulating strings, Using istream /ostream member functions, Using Manipulators, Creating Manipulator Functions, Understanding Implementation of Files, Writing and Reading Objects.

Templates: Generic Programming: and mastering STL Understanding Generic Functions with implementation of searching sorting algorithm, Overloading of Function Templates, Standard

Template Library: Understanding Components of Standard Template Library, Working of Containers, Algorithms, Iterators and Other STL Elements, Implementation of Sequence and Associative containers for different Algorithms using their Iterator.

References

1. Rumbaugh James et al, “Object Oriented Design and Modeling”, PHI-1997
2. Bjarne Stroustrup, “C++ Programming Language”, Addison Wesley
3. Balagurusamy E, “Object Oriented Programming with C++”, TMH, 2001
4. Booch Grady, “Object Oriented Analysis and Design with application 3/e”, Pearson
5. Lipman, Stanley B, Jonsce Lajole, “ C++ Primer Reading”, AWL, 1999
6. Dillon and Lee, “Object Oriented Conceptual Modeling”, New Delhi PHI-1993
7. Stephen R. Shah, “Introduction to Object Oriented Analysis and Design”, TMH
8. Berzin Joseph, “Data Abstraction: the object oriented approach using C++”, McGraw Hill
9. Mercer, “Computing Fundamental with C++”, Palgrave Macmillan

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER IV	MCA 104	OPERATING SYSTEMS	L	T	P	C
			3	1	3	4
GOAL	To provide thorough knowledge on the major functions of operating system.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. Learn the structure and function of operating systems 2. Learn the Process Scheduling and Deadlock processing. 3. Understand Device management. 4. Learn how memory management is done by the operating system. 5. Learn the concepts of I/O management and File systems management.			At the end of the course the student should be able to: 1. Explain the basic functions and structure of operating systems. 2. Implement the process scheduling algorithms and process synchronization techniques. 3. Implement Memory Management Techniques. 4. Illustrate File system and disk I/O techniques.			

UNIT I:

Introduction: Definition and types of operating systems, Batch Systems, multi programming, time-sharing parallel, distributed and real-time systems, Operating system structure, Operating system components and services, System calls, system programs, Virtual machines.

UNIT II:

Process Management: Process concept, Process scheduling, Cooperating processes, Threads, Interprocess communication, CPU scheduling criteria, Scheduling algorithms, Multiple-processor scheduling, Real-time scheduling and Algorithm evaluation.

UNIT III:

Process Synchronization and Deadlocks: The Critical-Section problem, synchronization hardware, Semaphores, Classical problems of synchronization, Critical regions, Monitors, Deadlocks-System model, Characterization, Deadlock prevention, Avoidance and Detection, Recovery from deadlock, Combined approach to deadlock handling.

UNIT IV:

Storage management: Memory Management-Logical and Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation with paging in MULTICS and Intel 386, Virtual Memory, Demand paging and its performance, Page replacement algorithms, Allocation of frames, Thrashing, Page Size and other considerations, Demand segmentation, File systems, secondary Storage Structure, File concept, access methods, directory implementation, Efficiency and performance, recovery, Disk structure, Disk scheduling methods, Disk management, Recovery, Disk structure, disk scheduling methods, Disk management, Swap-Space management, Disk reliability.

UNIT V:

Security & Case Study: Protection and Security-Goals of protection, Domain of protection, Access matrix, Implementation of access Matrix, Revocation of Access Rights, language based protection, The Security problem, Authentication, One Time passwords, Program threats, System threats, Threat Monitoring, Encryption.

Windows NT-Design principles, System components, Environmental subsystems, File system, Networking and program interface, Linux system-design principles, Kernel Modules, Process Management, Scheduling, Memory management, File Systems, Input and Output, Interprocess communication, Network structure, security

References

.

1. Abraham Siberschatz and Peter Baer Galvin, "Operating System Concepts", Addison-Wesley
2. Milan Milankovic, "Operating Systems, Concepts and Design", Tata McGraw-Hill.
3. Harvey M Deital, "Operating Systems", Addison Wesley
4. Richard Peterson, "Linux: The Complete Reference", Osborne Tata McGraw-Hill.
5. Sibsankar Halder and Alex A Aravind, "Operating Systems", Pearson Education
6. D M Dhamdhare, "Operating Systems : A Concept based Approach", McGraw Hill.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER V	MCA 105	PROGRAMMING WITH MATLAB	L	T	P	C
			3	1	3	4
GOAL	1. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. 2. It provides a gentle introduction to the MATLAB computing environment, and is intended for beginning users and those looking for a review.					
OBJECTIVES		OUTCOMES				
The course should enable the students to: 1. To introduce you to the basic operations of MATLAB 2. Understanding the MATLAB environment 3. Being able to do simple calculations using MATLAB 4. Being able to carry out simple numerical computations and analyses using MATLAB 5. Application development, including Graphical User Interface building. 6. It is designed to give students a intended for beginning users and those looking for a review.		At the end of the course the student should be able to: 1. Understand the main features of the MATLAB development environment 2. Use the MATLAB GUI effectively 3. Design simple algorithms to solve problems 4. Write simple programs in MATLAB to solve scientific and mathematical problems 5. Know where to find help 6. It is designed to give students a basic understanding of MATLAB, including popular toolboxes.				

UNIT I:

Introduction: Features, basics of MATLAB, MATLAB workspace, MATLAB desktop , Creating and running scripting M-files, MATLAB as a calculator, Variables, Comments, Complex numbers, Arithmetic operations with scalars, Floating-point arithmetic, Mathematical functions, Elementary math built-in functions, Commands for managing variables, Relational and Logical operations, Applications in problem solving.

UNIT II:

Arrays and Matrices : One dimensional array, Array addressing and indexing, Array manipulation, Array sorting, Construction and manipulation of multi-dimensional arrays, Built-in functions for handling arrays, Matrix manipulation, Array and matrix operations,. Character strings: String construction, string manipulation and string functions.

UNIT III:

Control Flow: The if, if-else construction, switch-case construction, for loop, while loop, nested loops. try-catch block.

Functions: Function construction, Rules for constructing functions, Input-output arguments, Executing functions, Sub-functions, Nested functions, Function handles and anonymous functions, Command line functions, Using a function file.

UNIT IV:

Two-dimensional graphics: Basic plots, plot function, Style options, Multiple plots, Multiple figures, Overlay plots, Specialized 2-D plots.

Three-dimensional graphics: Line plots, mesh plots, Surface plots, Contour plots, Changing viewpoints, Specialized 3-D plots.

UNIT V:

Applications: Linear algebra- solving linear system, Finding Eigen values and Eigen vectors; Polynomials- roots, addition, multiplication, division, curve fitting; data analysis; Differentiation, Integration, Solving differential equations.

References:

1. PratapR., “Getting Started with MATLAB- A Quick Introduction for Scientists and Engineers”, Oxford University Press.
2. Dukkupati R.V., “MATLAB- An Introduction with Applications”, New Age International Publishers.
3. HanselmanD. and LittlefieldB., “Mastering MATLAB 8”, Pearson Education.
4. Gilat A., “MATLAB: An Introduction with Applications”, John Wiley & Sons.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 106	CYBER SECURITY & INFORMATION SYSTEM	L	T	P	C
			3	1	3	4
GOAL	1. Three fundamental goals of cyber security are confidentiality, integrity and availability. 2. Goal of information System to help executives of an organization make decisions that advance the organization's strategy and to implement the organizational structure and dynamics of the enterprise for the purpose of managing the organization in a better way for a competitive advantage.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. cyber security prevent or mitigate harm to—or destruction of—computer networks, applications, devices, and data. 2. cyber security Boost innovation by ensuring safer collaboration across all environments, including cloud and mobile. 3. cyber security provide Scale intelligently and securely. 4. Information System should be based on a long-term planning. 5. Information System should provide a holistic view of the dynamics and the structure of the organization. 6. Information System should work as a complete and comprehensive system covering all interconnecting sub-systems within the organization. 7. Effectively communicate strategic alternatives to facilitate decision making.			At the end of the course the student should be able to: 1. To protect confidentiality of critical data. 2. To protect intellectual property in systems. 3. To test and mitigate security (e.g., in software) vulnerabilities. 4. To understand and be able to analyze post compromise. 5. For cyber security strategy to succeed, it must continually evolve to keep pace with the shifting strategies and technologies used by hackers. 6. Understand the leadership role of Management Information Systems in achieving business competitive advantage through informed decision making. 7. Analyze and synthesize business information and systems to facilitate evaluation of strategic alternatives..			

UNIT I:

Introduction to information systems, Types of information Systems, Development of Information Systems, Introduction to information security, Need for Information security, Threats to Information Systems, Information Assurance, Cyber Security, and Security Risk Analysis.

UNIT II:

Application security (Database, E-mail and Internet), Data Security Considerations-Backups, Archival Storage and Disposal of Data, Security Technology-Firewall and VPNs, Intrusion Detection, Access Control.

Security Threats -Viruses, Worms, Trojan Horse, Bombs, Trapdoors, Spoofs, E-mail viruses, Macro viruses, Malicious Software, Network and Denial of Services Attack, Security Threats to E-Commerce- Electronic Payment System, e-Cash, Credit/Debit Cards. Digital Signature, public Key Cryptography.

UNIT III:

Developing Secure Information Systems, Application Development Security, Information Security Governance & Risk Management, Security Architecture & Design Security Issues in Hardware, Data Storage & Downloadable Devices, Physical Security of IT Assets, Access Control, CCTV and intrusion Detection Systems, Backup Security Measures.

UNIT IV:

Security Policies, Why Policies should be developed, WWW policies, Email Security policies, Policy Review Process-Corporate policies-Sample Security Policies, Publishing and Notification Requirement of the Policies.

Information Security Standards-ISO, IT Act, Copyright Act, Patent Law, IPR. Cyber Laws in India; IT Act 2000 Provisions, Intellectual Property Law: Copy Right Law, Software License, Semiconductor Law and Patent Law.

UNIT V:

Business applications of information technology: Internet & electronic commerce, Intranet, Extranet & Enterprise Solutions, Information System for Business Operations, Information System for Managerial Decision Support, Information System for Strategic Advantage.

Concepts of planning & control: Concept of organizational planning, The Planning Process, Computational support for planning, Characteristics of control process, The nature of control in an organization.

Managing Information Technology: Enterprise & global management, Security & Ethical challenges, Planning & Implementing changes.

Advanced Concepts in Information Systems: Enterprise Resource Planning, Supply Chain Management, Customer Relationship Management, and Procurement Management.

References:-

1. Charles P. Pfleeger, Shari Lawerance Pfleeger, "Analysing Computer Security", Pearson Education India.
2. V.K. Pachghare, "Cryptography and information Security", PHI Learning Private Limited, Delhi India.
3. Dr. Surya Prakash Tripathi, Ritendra Goyal, Praveen kumar Shukla, "Introduction to Information Security and Cyber Law" Willey Dreamtech Press.
4. Schou, Shoemaker, "Information Assurance for the Enterprise", Tata McGraw Hill.
5. CHANDER, HARISH, "Cyber Laws And It Protection", PHI Learning Private Limited, Delhi, India
6. Brian, "Management Information System", Tata Mcgraw-hill Education Pvt. Ltd.
7. Gordon B. Davis & Margrethe H. Olson, "Management Information System", Tata Mcgraw-hill Education Pvt. Ltd.
8. Brian, "Introduction to Information System", Tata Mcgraw-hill Education Pvt. Ltd.
9. Davis, "Information System", Palgrave Macmillan.
10. Murdick, "Information System for Modern Management", PHI Learning Private Limited, Delhi India.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB I	MCA 107	PROGRAMMING IN ‘C’ & DATA STRUCTURE LAB	L	T	P	C
			0	0	3	2
GOAL	1. To develop the programming skill using C programming languages. 2. To provide an in-depth knowledge in problem solving techniques using data structures.					
OBJECTIVES		OUTCOMES				
The course should enable the students to: 1. Gain knowledge in C programming. 2. To learn problem solving in C. 3. Understand the concepts and applications of stack and queue data structures. 4. Learn various sorting and searching techniques. 5. Understand the concepts and applications of tree and graph data structures. 6. Have a good understanding of problem solving using data structure.		At the end of the course the student should be able to: 1. Develop applications using C programming 2. Write and execute the C programs for small applications. 3. Implement stacks and queues for various applications. 4. Implement tree data structure for different applications. 5. Implement various sorting and searching techniques. 6. Apply the concepts of graph for computing shortest path and construct MST.				

Suggested Assignments to be conducted on a 3-hour slot. It will be conducted in tandem with the theory course so the topics for problems given in the lab are already initiated in the theory class. The topics taught in the theory course should be appropriately be sequenced for synchronization with the laboratory. A sample sequence of topics and lab classes for the topic are given below:

Write Program in C for following:

1. Familiarization of a computer and the environment and execution of sample programs
2. Expression evaluation
3. Conditionals and branching
4. Iteration
5. Functions
6. Recursion
7. Arrays
8. Structures
9. Pointers
10. Sorting programs: Bubble sort, Merge sort, Insertion sort, Selection sort, and Quick sort.
11. Searching programs: Linear Search, Binary Search.
12. Array implementation of Stack, Queue, Circular Queue, Linked List.
13. Implementation of Stack, Queue, Circular Queue, Linked List using dynamic memory allocation.
14. Implementation of Binary tree.
15. Program for Tree Traversals (preorder, inorder, postorder).
16. Program for graph traversal (BFS, DFS).
17. Program for minimum cost spanning tree, shortest path.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB II	MCA 108	COMPUTER ORGANIZATION LAB	L	T	P	C
			0	0	3	2
GOAL	To learn the architecture programming and learn the basic methods and provide the fundamental concepts used in the design of digital systems.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. Outline the formal procedures for the analysis and design of combinational circuits, 2. Learn about several structural and behavioral models for synchronous sequential circuits, 3. Provide knowledge of the concept of memories and programmable logic devices.			At the end of the course the student should be able to: 1. Ability to understand and analyse, linear and digital electronic circuits. 2. To understand and apply computing platform and software for engineering problems. 3. Design and implement interfacing. 4. Use the design methodology for combinational logic circuits. 5. Make use of design concepts of sequential circuits. 5. Design and develop projects using microprocessor			

Objective: To understand the digital logic and create various systems by using these logics.

1. Introduction to digital electronics lab- nomenclature of digital ICs, specifications, study of the data sheet, concept of Vcc and ground, verification of the truth tables of logic gates using TTL ICs.

2. Implementation of the given Boolean function using logic gates in both SOP and POS forms.

3. Verification of state tables of RS, JK, T and D flip-flops using NAND & NOR gates.

4. Implementation and verification of Decoder/De-multiplexer and Encoder using logic gates.

5. Implementation of 4x1 multiplexer using logic gates.

6. Implementation of 4-bit parallel adder using 7483 IC.

7. Design, and verify the 4-bit synchronous counter.

8. Design, and verify the 4-bit asynchronous counter.

Microprocessor Experiment

1. Simple arithmetic operations: addition / subtraction / multiplication / division.

2. Programming with control instructions: (i) Ascending / Descending order, Maximum / Minimum of numbers (ii) Programs using Rotate instructions (iii) Hex / ASCII / BCD code conversions.

3. Interface Experiments: (i) A/D Interfacing. & D/A Interfacing

4. Traffic light controller.

5. I/O Port / Serial communication

6. Programming Practices with Simulators/Emulators/open source

7. Read a key board interface display (8279)

8. Demonstration of basic instructions with 8051 Micro controller execution, including: (i) Conditional jumps, looping 74 (ii) Calling subroutines.

9. Programming I/O Port 8051 (i) study on interface with A/D & D/A (ii) study on interface with DC & AC motor .

10. Mini project development with processors.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB III	MCA 109	OBJECT ORIENTED SYSTEM WITH C++ LAB	L	T	P	C
			0	0	3	2
GOAL	To practice the concepts learned in the subject Object Oriented Programming and C++.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. Gain knowledge in C++ programming. 2. To learn problem solving in C++. 3. Have a practical exposures in Object Oriented Programming (OOP). 4. Gain knowledge in object oriented concepts. 5. Work practically on day to day problems and to solve them using C++			At the end of the course the student should be able to: 1. Develop applications using C++ programming 2. Write and execute the C++ programs for small applications. 3. Develop solutions for a range of problems using objects and classes. 4. Demonstrate the implementation of constructors, destructors and operator overloading. 5. Apply fundamental algorithmic problems including type casting, inheritance, and polymorphism. 6. Write programs using generic programming, exception handling ,templates, file Handling 7. Analyze, design and develop solutions to real-world problems applying OOP Concepts of C++.			

Suggested Assignments to be conducted on a 3-hour slot. It will be conducted in tandem with the theory course so the topics for problems given in the lab are already initiated in the theory class. The topics taught in the theory course should be appropriately be sequenced for synchronization with the laboratory. A sample sequence of topics and lab classes for the topic are given below:

Write programs in C++ for

1. Program illustrating overloading of various operators.
2. Program illustrating use of Friend, Inline, Static Member functions, default arguments.
3. Program illustrating use of destructor and various types of constructor.
4. Program illustrating various forms of Inheritance.
5. Program illustrating use of virtual functions, virtual Base Class.
6. Program illustrating how exception handling is done.
7. Program implementing various kinds of Sorting programs: Bubble sort, Merge sort, Insertion sort, Selection sort, and Quick sort.
8. Program implementing various kinds of Searching programs: Linear Search, Binary Search.
9. Implementation of Binary tree.
10. Program for Tree Traversals (preorder, inorder, postorder).
11. Program for graph traversal (BFS, DFS).
12. Program for minimum cost spanning tree, shortest path.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB IV	MCA 110	OPERATING SYSTEMS LAB	L	T	P	C
			0	0	3	2
GOAL	To implement operating system(OS) concepts in LINUX platform and familiarise with low level system programming.					
OBJECTIVES			OUTCOMES			
The course should enable the students to:			At the end of the course the student should be able to:			
1. To understand and implement the basic resource management technique [Processor, Memory]			1. Simulate the principles of resource management [Processor, Memory].			
2. To solve the problems related with synchronization, concurrency related issues.			2. Install and use operating systems with an understanding of professional, ethical and social issues. [Windows, Linux etc.,]			
			3. Recognize the life long need and engage in upgradation of operating system			

LIST OF EXPERIMENTS

1. Program to report the behavior of the OS to get the CPU type and model, kernel version.
2. Shell programming
 - a. command syntax
 - b. write simple functions
 - c. basic tests
3. Shell programming
 - a. loops
 - b. patterns
 - c. expansions
 - d. substitution
4. Program to get the amount of memory configured into the computer, amount of memory currently available.
5. Implement the various process scheduling mechanisms such as FCFS, SJF, Priority, round – robin.
6. Implement the solution for reader – writer’s problem.
7. Implement the solution for dining philosopher’s problem.
8. Implement banker’s algorithm.
9. Implement the first fit; best fit and worst fit file allocation strategy.
10. Write a program to create processes and threads.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB V	MCA 111	PROGRAMMING WITH MATLAB	L	T	P	C
			0	0	3	2
GOAL	1. It integrates computation, visualization, and programming in an easy-to-use environment. It provides a gentle introduction to the MATLAB computing environment, and is intended for beginning users and those looking for a review.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. To introduce you to the basic operations of MATLAB 2. Understanding the MATLAB environment 3. Being able to do simple calculations using MATLAB 4. Being able to carry out simple numerical computations and analyses using MATLAB 5. Application development, including Graphical User Interface building. 6. It is designed to give students a intended for beginning users and those looking for a review.			At the end of the course the student should be able to: 1. Understand the main features of the MATLAB development environment 2. Use the MATLAB GUI effectively 3. Design simple algorithms to solve problems 4. Write simple programs in MATLAB to solve scientific and mathematical problems 5. Know where to find help 6. It is designed to give students a basic understanding of MATLAB, including popular toolboxes.			

LIST OF EXPERIMENTS

1. WAP to demonstrate the use of elementary math built-in functions.
2. WAP to create a menu-driven program using switch-case statement.
3. WAP to demonstrate the use of various arithmetic and relational operators.
4. WAP to perform basic operations on an array.
5. WAP to perform basic operations on a matrix.
6. WAP to demonstrate the use of if and if-else statement.
7. WAP to demonstrate the use of while loop.
8. WAP to demonstrate the use of for loop.
9. WAP to demonstrate the use of nested loops.
10. WAP to sort a vector in ascending order.
11. WAP to swap elements of a vector using function.
12. WAP to create and access strings.
13. WAP to plot a 2-D plot with given specifications.
14. WAP to plot a 3-D plot with given specifications.
15. WAP to solve a given differential equation.
16. WAP to perform differentiation of given equation.
17. WAP to perform integration of given equation.
18. WAP to solve a given polynomial.
19. WAP to find Eigen values and Eigen vector.
20. WAP to demonstrate try-catch block.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB VI	MCA 112	PROJECT / SEMINAR*	L	T	P	C
			0	0	3	2
GOAL	1. The project goal refers to achieving a desired outcome (Theory and Practical knowledge of MCA IST SEM on Specific paper) at a specific end date employing a specific amount of resources. 2. Seminar may be for the purpose of education, such as a lecture, where the participants engage in the discussion of an academic subject for the aim of gaining a better insight into the subject.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. A project objective is a statement that describes the “what” of your project. The tangible and measurable “what”. The “what” that's achievable, realistic, and can be completed within the time allowed. These statements ladder up to the goals of the project, providing stepping stones to success. What should knowledge gain from MCA IST SEM 2. To provide a comprehensive understanding of the Basic knowledge of the MCA IST SEM on specific Paper 3. To encourage the use of the knowledge by PPT 4. To promote positive relations between Faculty members and Classmates 5. To develop informed and responsibility			At the end of the course the student should be able to: 1. The value of any project cannot be measured without defining success. Outcomes are specific, measurable, and meaningful. knowledge gain from MCA IST SEM on specific Paper 2. Student Learning Outcomes from Seminar i. Presentation Skills. ii. Discussion Skills. iii. Listening Skills. iv. Argumentative Skills and Critical Thinking. Questioning. v. Interdisciplinary Inquiry. vi. Engaging with Big Questions. vii. Studying Major Works.			

*Seminar means giving explanation for a group of people. Paper presentation means presenting an explanation on a specified concept may include seminar in it.

*Project means doing for result oriented it may include both above two.

Students make a project thesis in specific topic which you read in this Semester. Maximum Two Students may be participated in one Project. After developing Project thesis student presenting an explanation on a specified concept may include seminar in it. Students develop project under the guidance of faculty members.

Prepare Project Thesis according Appendix 1

Dr. RamManohar Lohia Avadh University, Ayodhya U.P.

Study and Evaluation Scheme

MCA (Master of Computer Applications)

(Effective From Session 2020-21)

Year – I Semester – II

Sl. No.	Subject Code	Subject Name	Periods			Evaluation Scheme					Credit
			L	T	P	Sessional Exams			ESE	Subject Total	
						CT	TA	Total			
	THEORY SUBJECT										
1	MCA 201	Web Technology	3	1	0	30	20	50	100	150	04
2	MCA 202	Database Management System	3	1	0	30	20	50	100	150	04
3	MCA 203	Advanced Java Programming	3	1	0	30	20	50	100	150	04
4	MCA 204	Design & Analysis of Algorithms	3	1	0	30	20	50	100	150	04
5	MCA 205	Software Engineering	3	1	0	30	20	50	100	150	04
6	MCA 206	Elective – I	3	1	0	30	20	50	100	150	04
	Practical										
7	MCA 207	Web Technology Lab	0	0	3	10	10	20	30	50	02
8	MCA 208	Database Management System Lab	0	0	3	10	10	20	30	50	02
9	MCA 209	Advanced Java Programming Lab	0	0	3	10	10	20	30	50	02
10	MCA 210	Design & Analysis of Algorithms Lab	0	0	3	10	10	20	30	50	02
11	MCA 211	Software Engineering Lab	0	0	3	10	10	20	30	50	02
12	MCA 112	IOT Project / Seminar *	0	0	3	10	10	20	30	50	02
		Total	18	6	18	-	-	-	-	1200	36

Elective – I

MCA 206-(i)	Embedded System
MCA 206-(ii)	Soft Computing
MCA 206-(iii)	Knowledge Base System
MCA 206-(iv)	Computer Graphics and Multimedia
MCA 206-(v)	Social Mobile Analytics & Cloud
MCA 206-(vi)	Software Quality Assurance and Software Project Management

CT: Class Test

TA: Teacher Assessment

L/T/P: Lecture/Tutorial/ Practical

MSE: Mid Semester Examination

ESE: End Semester Examination

*Seminar means giving explanation for a group of people. Paper presentation means presenting an explanation on a specified concept may include seminar in it.

*Project means doing for result oriented it may include both above two.

Students make a project thesis in IOT based specific topic which you read in this Semester. Maximum Two Students may be participated in one Project. After developing Project thesis student presenting an explanation on a specified concept may include seminar in it. Students develop project under the guidance of faculty members.

End of MCA IIND Semester Exam students will do 45 days Industry Internship / training which is based on Latest technology for MCA IIIRD Semester Industry Project Viva

Prepare Project Thesis according MCA Thesis Guideline 2020

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER I	MCA 201	WEB TECHNOLOGY	L	T	P	C
			3	1	3	4
GOAL	To impart knowledge and training on HTML, XML, AJAX, ASP, JSP and web services.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Create a HTML and XML application using structure and presentation technologies. 2. Use XML manipulation technologies such as XSLT, XPath, XLink and XQuery. 3. Create a AJAX, ASP, JSP application using structure and presentation technologies. 4. Perform Program Manipulation and Dynamic access through DOM architecture. 5. Understand web services and ensure security. 6. Understand role of metadata in web content			The student should be able to 1. Apply HTML and XML concepts to develop Web application. 2. Create SOAP application using XML and Web Services. 3. Acquire information from the web sites using XML programming. 4. Apply AJAX, ASP and JSP concepts to develop Web application. 5. Implement XML in e-business 6. Develop web services and ensure security			

UNIT I:

Introduction: Introduction to web, protocols governing the web, web development strategies, Web applications, web project, web team.

Web Page Designing: HTML: list, table, images, frames, forms, CSS, Dynamic HTML.

UNIT II:

XML: DTD, XML schemes, Object Models, presenting and using XML,

Using XML Processors: DOM, SOPA and SAX, presentation technologies XSL, XFORMS, XHTML, voice XML, Transformation, XSLT, XLINK, XPATH, XQ.

Architecting Web Services: Business motivations for web services, B2B, B2C, Technical motivations, Service oriented Architecture (SOA), Architecting web services, Implementation view, web services technology stack, logical view, composition of web services, deployment view from application server to peer to peer process view , life in the runtime.

UNIT III:

Web Services Building Block: Transport protocols for web services, messaging with web services, protocols, SOAP, describing web services WSDL , Discovering web services UDDI, Securing web services.

XML and Content Management: Semantic Web, Role of Meta data in web content, Resource Description Framework, RDF schema, Architecture of semantic web, content management workflow, XLANG, WSFL.

UNIT IV:

Scripting: Java script: Introduction, documents, forms, statements, functions, objects; event and event handling; introduction to AJAX, VB Script

UNIT V:

Server Site Programming: Introduction to active server pages (ASP), ASP.NET, java server pages (JSP), JSP application design, tomcat server, JSP objects, declaring variables, and methods, debugging, sharing data between JSP pages, Session, Application: data base action , development of java beans in JSP, introduction to COM/DCOM/ CORBA

Text Books

1. Ron schmelzer et al, —XML and Web Services, Pearson Education, 2008.
2. Sandeep Chatterjee and James Webber, —Developing Enterprise Web Services: An Architect's Guide, Prentice Hall, 2004

References

- 1.Xavier, C, “ Web Technology and Design” , New Age International.
- 2.Ivan Bayross,” HTML, DHTML, Java Script, Perl & CGI”, BPB Publication.
- 3.Ramesh Bangia, “Internet and Web Design” , New Age International
- 4.Bhave, “Programming with Java”, Pearson Education
5. Ullman, “PHP for the Web: Visual QuickStart Guide”, Pearson Education
- 6.Deitel, “Java for programmers”, Pearson Education
6. Frank P. Coyle, —XML, Web Services and the Data Revolution, Pearson Education, 2002.
7. Keith Ballinger, —.NET Web Services Architecture and Implementation, Pearson Education, 2003.
8. Henry Bequet and Meeraj Kunnumpurath, —Beginning Java Web Services, Apress, 2004.
9. Russ Basiura and Mike Batongbacal, —Professional ASP.NET Web Services, Apress,2.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER II	MCA 202	DATABASE MANAGEMENT SYSTEM	L	T	P	C
			3	1	3	4
GOAL	To learn the fundamentals of data models and to conceptualize and depict a database system using ER diagram.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To make a study of SQL and relational database design. 2. To understand the internal storage structures using different file and indexing techniques which will help in physical DB design. 3. To know the fundamental concepts of transaction processing- concurrency control techniques and recovery procedure. 4. To have an introductory knowledge about the emerging trends in the area of distributed DB- OODB- Data mining and Data Warehousing. 5. To learn the basics of query evaluation and optimization techniques.			The student should be able to 1. Explore the basic concepts of database systems. 2. Write SQL queries for a given scenario. 3. Describe relational database theory, and be able to write relational algebra expressions for queries. 4. Design logical data models 5. Evaluate and optimize queries 6. Implement transaction processing and concurrency control 7. Develop Object oriented dB, Distributed dB using XML, data warehousing			

UNIT I:

Introduction: An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

UNIT II:

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

Introduction to SQL: Characteristics of SQL, Advantages of SQL, SQL data types and literals, Types of SQL commands, SQL operators and their procedure, Tables, views and indexes, Queries and sub queries, Aggregate functions, Insert, update and delete operations, Joins, Unions, Intersection, Minus, Cursors in SQL. PL/SQL, Triggers and clusters.

UNIT III:

Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependencies, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

UNIT IV:

Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi-version schemes, Recovery with concurrent transaction. Transaction Processing in Distributed system, data fragmentation, Replication and allocation techniques for distributed system, overview of concurrency control and recovery in distrusted database.

Unit- V

Data Storage And Query Processing: Record storage and Primary file organization, Secondary storage Devices, Operations on Files, Heap File, Sorted Files, Hashing Techniques , Index Structure for files, Different types of Indexes, B-Tree, B+Tree , Query Processing.

Object Oriented Databases: Need for Complex Data types, OO data Model, Nested relations, Complex Types, Inheritance Reference Types, Distributed databases, Homogenous and Heterogeneous, Distributed data Storage, XML, Structure of XML, XML Document Schema, Querying and Transformation, Data Mining and Data Warehousing.

Text Book

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan- —Database System Concepts, Sixth Edition, McGraw-Hill, 2010.

References

1. Date C J, “An Introduction To Database System”, Addison Wesley
2. Korth, Silbertz, Sudarshan, “Database Concepts”, Tata Mcgraw-hill Education (India) Pvt. Ltd.
3. Elmasri, Navathe, “Fundamentals Of Database Systems”, Pearson Education New Delhi India.
4. G.K. Gupta, “Database Management System”, Tata Mcgraw-hill Education (India) Pvt. Ltd.
5. Bipin C. Desai, “An introduction to Database Systems”, Galgotia Publication Pvt. Ltd. New Delhi.
6. Majumdar & Bhattacharya, “Database Management System”, Tata Mcgraw-hill Education Pvt. Ltd.
7. Ramakrishnan, Gehrke, “Database Management System”, McGraw Hill (India) Pvt Ltd. New Delhi.
8. Chakravarti, “Advanced Database Mnagement System” Wiley Dreamtech Publications.
9. Ramez Elmasri and Shamkant B. Navathe, —Fundamental Database Systems, Fifth Edition, Pearson Education, 2007.
10. Raghu Ramakrishnan, —Database Management System, Tata McGraw-Hill Publishing Company, 2003.
11. Hector Garcia–Molina, Jeffrey D.Ullman and Jennifer Widom- —Database System Implementation- Pearson Education- 2000.
12. Peter Rob and Corlos Coronel- —Database System, Design, Implementation and Management, Thompson Learning Course Technology- Fifth edition, 2003.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER III	MCA 203	ADVANCED JAVA PROGRAMMING	L	T	P	C
			3	1	3	4
GOAL	1. To impart training in JAVA programming skill development. 2. To create the awareness and comprehensive knowledge in Engineering Mathematics.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To understand the basic concepts of Java 2. To Understand Packages and Interfaces 3. To understand exception handling and Multi Threading in Java 4. Understand IO Streams 5. To understand the GUI part of Java 6. Understand Multi-tier concept in web application, creation and deployment of web application. 7. Understand programming using Servlet and JSP. 8. Understand Modular enterprise application development using EJB. 9. Understand the concept of struts framework. 10. Understand Database connection using Hibernate and creating web applications using Eclipse IDE			The student should be able to 1. Write basic program in Java 2. Create and Use packages and interfaces in java 3. Use user defined and inbuilt exceptions Create multi Threaded Applications 4. Use all types of Character and Byte Streams 5. Create GUI based trivial applications 6. Conceptually ready to create and deploy web applications. 7. Write web applications using Servlet and JSP. 8. Create simple business logic for enterprise applications using EJB. 9. Create simple enterprise application using struts framework 10. Create and deploy web applications using eclipse IDE and create Database connectivity using Hibernate.			

UNIT I:

Core Java: Introduction, Operator, Data type, Variable, Arrays, Control Statements, Methods & Classes, Inheritance, Package and Interface, Exception Handling, Multithread programming, I/O, Java Applet, String handling, Networking, Event handling, Introduction to AWT, AWT controls, Layout managers, Menus, Images, Graphics.

UNIT II:

Java Swing: Creating a Swing Applet and Application, Programming using Panes, Pluggable Look and feel, Labels, Text fields, Buttons, Toggle buttons, Checkboxes, Radio Buttons, View ports, Scroll Panes, Scroll Bars, Lists, Combo box, Progress Bar, Menus and Toolbars, Layered Panes, Tabbed Panes, Split Panes, Layouts, Windows, Dialog Boxes, Inner frame.

JDBC: The connectivity Model, JDBC/ODBC Bridge, java.sql package, connectivity to remote database, navigating through multiple rows retrieved from a database.

UNIT III:

Java Beans: Application Builder tools, The bean developer kit(BDK), JAR files, Introspection, Developing a simple bean, using Bound properties, The Java Beans API, Session Beans, Entity Beans, Introduction to Enterprise Java beans (EJB),

UNIT IV:

Introduction to RMI (Remote Method Invocation): A simple client-server application using RMI.

Java Servlets: Servlet basics, Servlet API basic, Life cycle of a Servlet, Running Servlet, Debugging Servlets, Thread-safe Servlets, HTTP Redirects, Cookies, Introduction to Java Server pages (JSP).

UNIT V:

Advanced Java: Java Transaction API (JTA), Java Messaging Services (JMS), AJAX, JINDI STRUTS, **Struts Framework:** Basics & Architecture, Request Handling Life Cycle, Building a simple struts, Configuration, Actions, Interceptors, Results, Value Stack/OGNL Struts2 Tag Libraries Struts2 XML Based Validations , Database Access.

Hibernate And IDE: Introduction to Hibernate, ORM Overview, Hibernate Environment , Hibernate Architecture & API, Hibernate Configuration, Hibernate Sessions, Persistent Class & Mapping Files, Building Hibernate application, Hibernate Query Language (HQL), Hibernate O/R Mappings, Collection & Association Mappings, Hibernate Annotations Eclipse, overview ,installation, perspectives, workspaces creating projects, packages, classes, interfaces, java build path, run Configuration, running program, creating JAR files, debugging programs and configurations, installing plugins developing application using Eclipse.

Text Books

1. Cay S. Horstman and Gary Cornell, —Core Java Volume I—Fundamentals, 9th Ed (Core Series), Prentice Hall, 2013
2. Herbert Schildt, —Java 2: The Complete Reference, 5th Ed, Tata McGraw Hill, 2002.

References:

1. Margaret Levine Young, “The Complete Reference Internet”, Tata McGraw-hill Education Pvt. Ltd.
2. Thampi, “Object Oriented Programming in JAVA” Wiley Dreamtech Publication.
3. Balagurusamy E, “Programming in JAVA”, Tata McGraw-hill Education Pvt. Ltd.
4. Dustin R. Callway, “Inside Servlets”, Addison Wesley.
5. Mark Wutica, “Java Enterprise Edition”, QUE.
6. Steven Holzner, “Java2 Black book”, Wiley Dreamtech Publication.
7. Liang, “Introduction to Java Programming, Comprehensive Version”, Pearson Education.
8. Deitel and Deitel, “Java: How to Program” PHI Learning Private Limited, Delhi India.
9. Jim Keogh, —The Complete Reference J2EE, Tata McGraw – Hill Edition 2002.
10. James Holmes, —The Complete References Struts, 2nd Edition, Tata McGraw, 2007.
11. http://www.idt.mdh.se/kurser/cd5480/2003/lectures/j2ee1_3tutorial.pdf
12. <http://www.tutorialspoint.com/eclipse/index.htm>
13. Jatin Couch, Daniel H. Steinberg, —J2EE Bible, Wiley India (P) Ltd, New Delhi 2002.
13. Paul Tremblett, —Instant Enterprise Java Y-Beans, Tata McGraw Hill, 2001.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER IV	MCA 204	DESIGN & ANALYSIS OF ALGORITHMS	L	T	P	C
			3	1	3	4
GOAL	The provide students with solid foundations to deal with a wide variety of computational problems, and to provide a thorough knowledge of the most common algorithms and data structures.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Learn asymptotic performance of algorithms. 2. Learn the algorithm analysis techniques.. 3. Become familiar with the different algorithm design techniques.. 4. Understand the limitations of Algorithm power 5. Understand the efficient algorithms			The student should be able to 1. Apply knowledge of computing and mathematics to algorithm design; 2. Analyze the time and space complexity of algorithms. 3. Design algorithms for various computing problems. 4. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them. 5. Critically analyze the different algorithm design techniques for a given problem.			

UNIT I:

Introduction: Algorithms, Analysis of Algorithms, Design of Algorithms, Complexity of Algorithms, Asymptotic Notations, Growth of function, Recurrences and their solution methods.

Sorting in polynomial Time: Insertion sort, Merge sort, Heap sort, and Quick sort

Sorting in Linear Time: Counting sort, Radix Sort, Bucket Sort, Medians and order statistics

UNIT II:

Advanced Data Structure: Red Black Trees, Augmenting Data Structure, Binomial Heap, B-Tree, Fibonacci Heap, and Data Structure for Disjoint Sets, All kinds of Algorithms on these data structures, Dictionaries and priority Queues, mergeable heaps, concatenable queues

UNIT III:

Advanced Design and Analysis Techniques: Dynamic programming, Greedy Algorithm, Backtracking, Branch-and-Bound, Amortized Analysis

UNIT IV:

Graph Algorithms: Elementary Graph Algorithms, Breadth First Search, Depth First Search, Minimum Spanning Tree, Kruskal's Algorithms, Prim's Algorithms, Single Source Shortest Path, All pair Shortest Path, Maximum flow and Traveling Salesman Problem

UNIT V:

Randomized Algorithms, String Matching, NP-Hard and NP-Completeness, Approximation Algorithms, Sorting Network, Matrix Operations, Polynomials and FFT, Number Theoretic Algorithms

Text Book:

1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education, 2007

References

1. Thomas H Cormen Leiserson “Introduction to Algorithms”, PHI Learning Private Limited, Delhi India.
2. Sara Baase and Allen Van Gelder ,Computer Algorithms : “Introduction to Design and Analysis”, Pearson Education
3. Jon Kleinberg and Eva Tardos “Algorithm Design”, Pearson Education
4. Brassard Bratley “Fundamental of Algorithms”, PHI Learning Private Limited, Delhi India.
5. M T Goodrich “Algorithms Design”, John Wiley
6. Aho, “Design and Analysis of Computer Algorithms”, Pearson Education.
7. Horowitz and Sahani ,“Fundamentals of Computer Algorithms”, Galgotia Publications Pvt Ltd Delhi India.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER V	MCA 205	SOFTWARE ENGINEERING	L	T	P	C
			3	1	3	4
GOAL	The main goal of this course is to impart knowledge on the basic principles of software development life cycle, testing and configuration management processes.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Understand the software life cycle models 2. Learn Requirement analysis and design concepts 3. Learn various software construction technologies. 4. Acquire knowledge on testing. 5. Understand the importance of SCM and release management			The student should be able to 1. Apply the concepts of life cycle models to choose the appropriate model. 2. Analyse the requirements and design the software. 3. Construct or implement the software based on the industry standards 4. Design and develop test cases 5. Work with version control and work on configuration and release management plans			

UNIT I:

Introduction: Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Overview of modern software engineering processes and practices(SCRUM,ADP), Software Quality Attributes. Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.

UNIT II:

Software Requirement Specifications (SRS): Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modeling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS. Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.

UNIT III:

Software Design: Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halstead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

UNIT IV:

Software Testing: Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.

Software Maintenance and Software Project Management: Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re- Engineering, Reverse Engineering. Software

Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

UNIT V:

Software Construction: Software construction fundamentals, Managing construction, Construction technologies, Construction tools, Structured coding Techniques, Coding Styles, Standards and Guidelines, Documentation Guidelines, Modern Programming Language Features: Type checking, User defined data types, Data Abstraction, Exception Handling, Concurrency Mechanism.

Software Configuration Management: SCM, Need for SCM, Version control, Introduction to SCM process, Software configuration items Management of the SCM process, Software configuration identification, Software configuration control Software configuration status accounting, Software configuration auditing, Software release management and delivery, Software configuration management tools

Text Book

1. Roger S.Pressman, Software engineering- A practitioner's Approach, 7thEd., McGraw-Hill, 2014.

References:

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
3. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
4. Pankaj Jalote, Software Engineering, Wiley
5. Carlo Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication.
6. Ian Sommerville, Software Engineering, Addison Wesley.
7. KassemSaleh,"Software Engineering", Cengage Learning.
8. Pfleeger, Software Engineering, Macmillan Publication.
- 9.. James F Peters and WitoldPedryez, —Software Engineering – An Engineering Approachll, John Wiley and Sons, New Delhi, 2000.
- 10.. Ali Behforooz and Frederick J Hudson, —Software Engineering Fundamentalsll, Oxford University Press, New Delhi, 1996.
- 11.. Sheikh Umar Farooq, S. M. K Quadri and Nesar Ahmad, —Software Testing Techniques Evaluation – An Empirical Approachll, Lambert Academic Publishing, Germany, Dec 2012 (ISBN: 978-3-659-19538-9).

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 206 (i)	Elective – I EMBEDDED SYSTEM	L	T	P	C
			3	1	3	4
GOAL	To provide basic knowledge about embedded systems design and understand the RTOS concepts					
OBJECTIVES		OUTCOMES				
The course should enable the students to 1. Understand the embedded systems hardware and software, 2. Understand the devices and buses used for embedded networking, 3. Understand the programming concepts and embedded programming in C and C++, 4. Understand the real time operating system concepts and inter-task communication, 5. Understand the Vx Works RTOS functions		The student should be able to 1. Know the concepts of embedded processors hardware, software and System on a Chip, 2. Know about the embedded interfacing devices, buses and networking protocols, 3. Know the embedded programming concepts in C and C++, 4. Know the concepts of real time operating system, inter process communication and synchronization, 5. Know the Vx Works RTOS concepts and functions.				

UNIT I:

Introduction to embedded systems: Definition and Classification, Characteristics and requirements, Applications, Overview of Processors and hardware units in an embedded system, Software embedded into the system, Exemplary Embedded Systems, Embedded Systems on a Chip (SoC) and the use of VLSI designed circuits.

UNIT II:

Devices And Buses For Devices: NETWORK I/O Devices, Device I/O Types and Examples, Synchronous, ISO synchronous and Asynchronous Communications from Serial Devices, Examples of Internal Serial Communication Devices - UART and HDLC, Parallel Port Devices, Sophisticated interfacing features in Devices/ Ports, Timer and Counting Devices - 12C, USB, CAN, and advanced I/O Serial high speed buses- ISA, PCI, PCI-X, CPCI and advanced buses, Timing and clocks in Embedded systems, Task Modeling and management, Real time operating system issues.

UNIT III:

Signals, frequency spectrum and sampling, digitization (ADC, DAC), Signal Conditioning and Processing. Modeling and Characterization of Embedded Computation System.

Programming Concepts And Embedded Programming In C, C++ : Programming in assembly language (ALP) vs. High Level Language, C Program Elements, Macros and functions ,Use of Pointers, NULL Pointers, Use of Function Calls, Multiple function calls in a Cyclic Order in the Main Function Pointers , Function Queues and Interrupt Service Routines Queues Pointers, Concepts of EMBEDDED PROGRAMMING in C++, Objected Oriented Programming, Embedded Programming in C++, C Program compilers, Cross compiler, Optimization of memory codes.

UNIT IV:

Embedded Control and Control Hierarchy, Communication strategies for embedded systems: Encoding and Flow control. Fault-Tolerance, FormalVerification , Trends in Embedded Processor, OS, Development Language

UNIT V:

REAL TIME OPERATING SYSTEMS: Definitions of process, tasks and threads , Clear cut distinction between functions, ISRs and tasks by their characteristics, Operating System Services, Goals, Structures, Kernel, Process Management, Memory Management, Device Management, File

System Organization and Implementation, I/O Subsystems, Interrupt Routines Handling in RTOS, Real Time Operating Systems : RTOS Task scheduling models, Handling of task scheduling and latency and deadlines as performance metrics, Co-operative Round Robin Scheduling, Cyclic Scheduling with Time Slicing (Rate Monotonics Co-operative Scheduling), Preemptive Scheduling Model strategy by a Scheduler, Critical Section Service by a Preemptive Scheduler, Fixed (Static) Real time scheduling of tasks, Inter Process Communication and Synchronisation, Shared data problem, Use of Semaphore(s) , Priority Inversion Problem and Deadlock Situations, Inter Process Communications using Signals, Semaphore Flag or mutex as Resource key, Message Queues, Mailboxes, Pipes ,Virtual (Logical) Sockets , Remote Procedure Calls (RPCs).

Textbook

1. Rajkamal, Embedded Systems Architecture, Programming and Design, TATA McGraw-Hill, Second Edition-2009.

References:

1. Steve Heath, Embedded Systems Design, Second Edition-2003, Newnes,
2. David E.Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint 2000.
3. Wayne Wolf, Computers as Components; Principles of Embedded Computing System Design – Harcourt India, Morgan Kaufman Publishers, First Indian Reprint 2001
4. Frank Vahid and Tony Givargis, Embedded Systems Design – A unified Hardware /Software Introduction, John Wiley, 2002.
5. Prasad, Embedded /Real Time System,Concept,Design and Programming Black Book, Wiley India
6. R.Gupta, “Co-synthesis of Hardware and Software for Embedded Systems”, Kluwer
7. Shibu K.V., “Introduction to Embedded Systems”, TMH
8. Marwedel, “Embedded System Design”, Springer

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 206 (ii)	Elective – I SOFT COMPUTING	L	T	P	C
			3	1	3	4
GOAL	To provide knowledge about soft computing techniques.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Learn the concepts of fuzzy sets, fuzzy logic and heuristics based on human experience. 2. Understand neural network techniques. 3. Learn the mathematical background for carrying out optimization associated with neural network learning. 4. Learn genetic algorithms and random search procedures. 5. Learn the methodologies to illustrate the intelligent behaviour of programs based on soft computing			The student should be able to 1. Apply concepts of fuzzy sets, fuzzy logic and heuristics based systems. 2. Derive appropriate rules for inference systems. 3. Have understood the mathematical background to optimize neural network learning. 4. Implement optimization algorithms and random search procedures useful to seek global optimum in self-learning. 5. Develop case studies to illustrate the intelligent behaviour of programs based on soft computing			

UNIT I:

Fuzzy Set Theory: Introduction to Neuro, Fuzzy and Soft Computing Fuzzy Sets , Basic Definition and Terminology , Fuzzy sets and Crisp sets, theoretic Operations , Member Function Formulation and parameterization , Fuzzy Rules and Fuzzy Reasoning , Extension Principle and Fuzzy Relations , Fuzzy If, Then Rules , Fuzzy Reasoning , Fuzzy Inference Systems , Mamdani Fuzzy Models , Sugeno Fuzzy Models , Tsukamoto Fuzzy Models , Input Space Partitioning and Fuzzy Modeling , Fuzzyfications & Defuzzificataions, Fuzzy Controller, Industrial applications

UNIT II:

Optimization: Derivative, based Optimization , Descent Methods , The Method of Steepest Descent , Classical Newton's Method , Step Size Determination , Derivative, free Optimization , Genetic Algorithms , Simulated Annealing , Random Search , Downhill Simplex Search, Particle Swarm Techniques , Ant Colony Optimization.

UNIT III:

Neural Networks:

Neuron, Nerve structure and synapse, Artificial Neuron and its model, activation functions, Neural network architecture: single layer and multilayer feed forward networks, recurrent networks, Supervised Learning Neural Networks , Various learning techniques; perception and convergence rule, Mutilayer Perceptrons , Mutilayer Perceptrons , Auto-associative and hetro-associative memory. Adaline , Back propagation, Radial Basis Function Networks , Unsupervised Learning Neural Networks , Competitive Learning Networks , Kohonen Self, Organizing Networks , Learning Vector Quantization , Hebbian Learning.

UNIT IV:

Neuro Fuzzy Modeling: Adaptive Neuro, Fuzzy Inference Systems , Architecture , Hybrid Learning Algorithm , Learning Methods that Cross-fertilize ANFIS and RBFN , Coactive Neuro Fuzzy Modeling , Framework Neuron Functions for Adaptive Networks , Neuro Fuzzy Spectrum.

UNIT V:

Applications of Computational Intelligence: Printed Character Recognition , Inverse Kinematics Problems , Automobile Fuel Efficiency Prediction , Soft Computing for Color Recipe Prediction.

Genetic Algorithm(GA) Basic concepts, working principle, procedures of GA, flow chart of GA, Genetic representations, (encoding) Initialization and selection, Genetic operators, Mutation, Generational Cycle, applications.

TEXT BOOK

1. J.S.R.Jang, C.T.Sun and E.Mizutani, —Neuro-Fuzzy and Soft Computing, Pearson, 2004.
2. N.P.Padhy, "Artificial Intelligence and Intelligent Systems" Oxford University Press.

REFERENCE BOOKS

1. Timothy J.Ross, —Fuzzy Logic with Engineering Applications, McGraw-Hill, 1997.
2. Davis E.Goldberg, —Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley, 2009.
3. S. Rajasekaran and G.A.V.Pai, —Neural Networks, Fuzzy Logic and Genetic Algorithms, PHI, 2003.
4. Siman Haykin, "Neural Networks" Prentice Hall of India
4. Timothy J. Ross, "Fuzzy Logic with Engineering Applications" Wiley India.
5. Kumar Satish, "Neural Networks" Tata Mc Graw Hill

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 206 (iii)	Elective – I KNOWLEDGE BASE SYSTEM	L	T	P	C
			3	1	3	4
GOAL	To capture the knowledge of human experts to support decision-making.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Learn about designing an infrastructure for AI 2. Learn about the impact of new AI chips on CIOs 3. Gain insight into how AI fits into business strategies 4. Learn more about the differences between case-based and rule-based knowledge-based systems 5. Learn Read more about the different aspects of knowledge-based systems. 6. Enables the users to query the knoweledge based system.			The student should be able to 1. The KBS employed in applications as diverse as avalanche path analysis, industrial equipment fault diagnosis and cash management. 2. To develop the architecture of the system – that it represents knowledge explicitly, rather than as procedural code. 3. It develop information in a given field -- medical diagnosis, scientific research 4. It develop an interface through which users query the system and interact with it. 5. Solutions to old problems represented cases			

UNIT I:

Basic concepts, Definition of AI, Background and past achievements, Aims Overview of application areas, 1.3 Problems and problem solving, State space search; Production rules; Logic Heuristic search techniques, Generate and test; Hill climbing; Search reduction strategies

UNIT II:

Representation models, Predicate logic; rules; Semantic nets; Frames; Conceptual graphs; Scripts, Fuzziness and uncertainty Security considerations, Fuzzy logic; Statistical techniques for determining probability, Methodologies for developing knowledge based systems, The KBS Development Life Cycle; Knowledge acquisition/elicitation. Management of KBS projects, Prototyping; Implementation; Development environments

UNIT III:

Neural networks, Architectures; Hopfield network; Multi-layer perception, Feed forward; Back propagation , Genetic algorithms, Basic concepts; Population; Chromosomes; Operators; Schemata; Coding , Rule induction, Basic concepts; Decision trees/rule sets

UNIT IV:

Expert systems, Natural language processing, Machine vision and robotics, Data mining and intelligent business support, Internet based application

UNIT V:

Reasoning with Uncertainty, Probability, Bayesian Decision Making, Dempster-Shafer Theory, Approximate Reasoning, Fuzzy Logic, Semantic Web Technologies

Textbooks:

1. Knowledge-Based Systems. Rajendra Akerkar , Priti Sajja, 2009 , ISBN10: 0763776475.
2. Engineering of Knowledge-Based Systems. Avelino J. Gonzalez, Douglas D. Dankel, Prentice Hall (2000), ISBN-10: 0130189731.
3. Expert Systems: Principles and Programming, Fourth Edition. Joseph C. Giarratano, Gary D. Riley, 2004, ISBN-10: 0534384471
4. Peter Jackson, Introduction to Expert Systems, Addison-Wesley (3rd Ed), 1998

5. Goldberg D. E., Genetic Algorithms in Search, Optimisation and Machine Learning, AddisonWesley, 1989.
6. Michalski, Bratko, Kubat, Machine Learning and Data Mining, Wiley (3rd Ed), 1999.
7. Expert Systems: Principles and Programming (Hardcover) Publisher: Course Technology; 4Rev Ed edition (15 Oct 2004).
8. A Bradford, Knowledge Engineering and Management: The Common KADS Methodology, 2000. MIT Press

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 206 (iv)	Elective – I COMPUTER GRAPHICS AND MULTIMEDIA	L	T	P	C
			3	1	3	4
GOAL	To provide the fundamental knowledge to develop interactive graphics and handling multimedia devices.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Learn the rules and algorithms in generating graphical outputs. 2. Learn 3-dimensional objects using suitable transformations. 3. Understand the architecture for design of multimedia system. 4. Realize the issues related to multimedia file handling. 5. Understand hypermedia standards in developing multimedia applications.			The student should be able to 1. Develop algorithms to draw fundamental drawings 2. Develop real-time rendering graphics 3. Create 2D and 3D images 4. Have an understanding on the basics of creating multimedia applications 5. Design and Develop multimedia applications			

UNIT I:

Introduction and Line Generation: Types of computer graphics, Graphic Displays- Random scan displays, Raster scan displays, Frame buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms, Mid point circle generating algorithm, and parallel version of these algorithms.

UNIT II:

Transformations: Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections and shearing. Windowing and Clipping: Viewing pipeline, Viewing transformations, 2-D Clipping algorithms- Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against non rectangular clip windows; Polygon clipping – Sutherland Hodgeman polygon clipping, Weiler and Atherton polygon clipping, Curve clipping, Text clipping.

UNIT III:

Three Dimensional: 3-D geometric primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping, Curves and Surfaces: Quadric surfaces, Spheres, Ellipsoid, Blobby objects, Introductory concepts of Spline, Bspline and Bezier curves and surfaces. Hidden Lines and Surfaces: Back Face Detection algorithm, Depth buffer method, A- buffer method, Scan line method, basic illumination models – Ambient light, Diffuse reflection, Specular reflection and Phong model, Combined approach, Warn model, Intensity Attenuation, Color consideration, Transparency and Shadows.

UNIT IV:

Computer Animations : Conventional and computer assisted animation, design of animation sequences, interpolation, simple animation effects, animation languages (Key Frame System, Parameterized systems), motion specifications, methods of controlling animation.

Multimedia Systems Design: An Introduction, Multimedia applications, Multimedia System Architecture, Evolving technologies for Multimedia, Defining objects for Multimedia systems, Multimedia Data interface standards, Multimedia Databases.

UNIT V:

Multimedia File Handling: Compression & Decompression, Data & File Format standards, Multimedia I/O technologies - Digital voice and audio, Video image and animation, Full motion video, Storage and retrieval Technologies.

Hypermedia: Multimedia Authoring & User Interface, Hypermedia messaging - Mobile Messaging, Hypermedia message component, Creating Hypermedia message, Integrated multimedia message standards, Integrated Document management, Distributed Multimedia Systems

Text Books

1. Prabat K Andleigh and Kiran Thakrar, —Multimedia Systems and Design, PHI, 2003 .

References:

1. Donald Hearn and M Pauline Baker, “Computer Graphics C Version”, Pearson Education
2. Amrendra N Sinha and Arun D Udai,” Computer Graphics”, TMH
3. Donald Hearn and M Pauline Baker, “ Computer Graphics with OpenGL”, Pearson education
4. Steven Harrington, “Computer Graphics: A Programming Approach”, PHI or TMH
5. James D Foley, A V Dam, S K Feiner and John f Hughes, “Computer Graphics Principles and Practice” Second Edition in C.
6. Judith Jeffcoate, —Multimedia in practice technology and Applications, PHI, 1998

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 206 (v)	Elective – I SOCIAL MOBILE ANALYTICS & CLOUD	L	T	P	C
			3	1	3	4
GOAL	The Objective is to familiarize students with the latest trends and advancements in Social Mobile Analytics and Cloud (SMAC)					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To familiarize students with the concepts of SMAC 2. To know the principles behind social media 3. To develop mobile application 4. To study about data analytics 5. To learn about the cloud services/models and benefits of cloud			The student should be able to 1. Have knowledge of SMAC 2. Familiarize with all the social media components 3. Design and develop mobile apps and appreciate the advantage of social media 4. Know the tools and use of data analytics 4. Understand various cloud service and deployment models			

UNIT I:

Introduction to SMAC: Introduction to SMAC, Interesting Facts, Objectives, SMAC - Social, SMAC Mobile, SMAC Analytics, SMAC Cloud, Business Benefits of Each Component Recent Market Trends, ADAPT, SMAC Challenges and Solution 3:5

UNIT II:

Social Media: Objectives, Social Media Overview, Online Networking, Social Media Marketing, Impact of Social Media on Business-Emerging Trends and Challenges in Social Media-Future of Social Media , Case Studies , Social Media , Opportunities

UNIT III:

Mobile Technology: Objectives, The Mobile Market, Mobile Technology, Growth and Reach, Mobile Platforms and Applications , Mobile Apps Market, Impact of Mobile, Mobility, Enterprise Mobility, Bring our Own Device (BYOD), Mobile Commerce, Mobile Payment, Mobile Wallets, Threats to Mobile Data , Consequences of Data Breaches, Securing Mobile Data, Machine to Machine , M2M in Healthcare, Automobile, Mobile in SMAC

UNIT IV:

Analytics: Objectives, Characteristics of Big Data, Digital Footprint, Categories of Digital Footprint, Big Data Analytics, Analytics in Industries, Emerging Trends and Challenges in Analytics-1 Future of Analytics, Case Studies , Analytics, Opportunities

UNIT V:

Cloud Computing: Cloud Computing Overview, Cloud Deployment Models, Cloud Service Models, Impact of Cloud Computing on Business, Emerging Trends and Challenges in Cloud, Future of Cloud Computing, Case Studies , Cloud Computing , Opportunities.

REFERENCE BOOKS

1. Gautam Shroff, —Enterprise Cloud Computing, Cambridge, 2010.
2. Scott Granneman, —Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop, Pearson Education, 2008.
3. Barry Burd, —Android Application Development All in one for Dummies, Edition I
4. Michael Berthold, David J. Hand, —Intelligent Data Analysis, Springer, 2007.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 206 (vi)	Elective – I SOFTWARE QUALITY ASSURANCE AND SOFTWARE PROJECT MANAGEMENT	L	T	P	C
			3	1	3	4
GOAL	To introduce an integrated approach to software development					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Learn the concepts of quality management. 2. Understand what quality standard is. 3. Study the Pareto principles and quality techniques. 4. Understand Software audit and cost. 5. Study the software quality assurance metrics.			The student should be able to 1. Have gained the knowledge in quality management, program and experience from experts. 2. Have obtained understanding in standards and training. 3. Have learnt the pareto principles and quality techniques. 4. Have grasped the concepts of software audit and cost of software quality system. 5. Be confident in software quality.			

UNIT I:

Software Quality In Business Context: Software Quality in Business Context : The meaning of Quality, The quality challenge, Why is Quality important, Quality control vs. Quality Assurance at each phase of SLDC, Quality Assurance in Software Support projects, The SQA function (Nina. S. Godbole).

Software Quality Assurance: Quality Concepts, Quality control, Quality Assurance, Cost of Quality. Software Quality Assurance Background issues, SQA Activities, Software Reviews, Cost impact of Software Defects, Defect Amplification and removal. Formal Technical reviews- The review meeting, Review reporting and Record keeping, Review guidelines, Sample-driven reviews, Formal Approaches to SQA, Software Reliability, Measures of Reliability and Availability, Software Safety, The SQA plan.(Pressman).

UNIT II:

Product Quality And Process Quality: Product Quality and process Quality: Introduction, Software Systems evolution, Product Quality, Models for product Quality, Process Quality, Software Measurement and Metrics: Introduction, Measurement during Software Life cycle Context, Defect Metrics, Metrics for Software Maintenance, Classification of Software Metrics, Requirements related metrics, Measurements and Process Improvement, Measurement principles, Identifying appropriate Measures and Metrics for Projects, Metrics implementation in projects, Earned Value Analysis, Issues in Software Measurements and Metrics program implementation, Object- Oriented Metrics: An Overview (Godbole).

UNIT III:

ISO 9000 FAMILY AND SIX SIGMA: ISO 9001: ISO 9000, ISO Standards Development Process. ISO 9000 family of standards, ISO 9001:2000, ISO Certification Surveillance Audits/RE-certification/Re-Assessment Audits. Six Sigma- Introduction, Six sigma in statistical context.

UNIT IV:

CMM AND PROCESS IMPROVEMENT MODELS: Software CMM and other process improvement models: CMM for software- an overview. Practices followed at mature organizations, Types of CMMs, CMM-Integrated model – What is CMM-I, Background to the CMNMODEL, Types of CMM-I models, Other models for Software Process Improvement and excellence – ISO 12207,

IEEE 1074, Malcom Baldrige National Quality Award, The EFQM Excellence Model, People – CMM.

UNIT V SOFTWARE TESTING: Software Testing – Overview, Purpose of Testing, Differences between Inspection and Testing, Testing vs Debugging, Testing Life Cycle, Test Artifacts, The Test Plan, The V-Model for testing Phases, Testing Techniques – Equivalence partitioning, Boundary value Analysis, State Transition Analysis, GUI Testing, Performance Testing, Reliability Testing, Risk-based testing. Gray Box Testing, Extreme testing, Test process improvement framework.

Text Book

1. Nina S Godbole, Software Quality Assurance: Principles and Practice, Narosa Publishing House, 2004.

Reference

1. Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Ed., McGraw-Hill, 2005.
2. Alka Jarvis and Vern Crandall, —In Roads to Software Quality: How to guide and toolkit, Prentice-Hall, 1997.
3. Pankaj Jalote, —Software Engineering Principles, Narosa Publishing House, 2000.
4. Richard Fairley, —Software Engineering concepts, Tata McGraw-Hill, 2001.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB I	MCA 207	WEB TECHNOLOGY LAB	L	T	P	C
			0	0	3	2
GOAL	To impart knowledge and training on HTML, XML, AJAX, ASP, JSP and web services.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Create a HTML and XML application using structure and presentation technologies. 2. Use XML manipulation technologies such as XSLT, XPath, XLink and XQuery. 3. Create a AJAX, ASP, JSP application using structure and presentation technologies. 4. Perform Program Manipulation and Dynamic access through DOM architecture. 5. Understand web services and ensure security. 6. Understand role of metadata in web content			The student should be able to 1. Apply HTML and XML concepts to develop Web application. 2. Create SOAP application using XML and Web Services. 3. Acquire information from the web sites using XML programming. 4. Apply AJAX, ASP and JSP concepts to develop Web application. 5. Implement XML in e-business 6. Develop web services and ensure security			

Objectives:-

1. Write HTML/Java scripts to display your CV in navigator, your Institute website, Department Website and Tutorial website for specific subject
2. Design HTML form for keeping student record and validate it using Java script.
3. Write an HTML program to design an entry form of student details and send it to store at database server like SQL, Oracle or MS Access.
4. Write programs using Java script for Web Page to display browsers information.
5. Write a Java applet to display the Application Program screen i.e. calculator and other.
6. Writing program in XML for creation of DTD, which specifies set of rules. Create a style sheet in CSS/ XSL & display the document in internet explorer.
7. Using ASP for server side programming, ASP for user name and password and to retrieve & match the value. It display success and failure messages. ASP for creating text file local drive, ASP for keeping the student record in database.
8. Program to illustrate JDBC connectivity. Program for maintaining database by sending queries. Design and implement a simple servlet book query with the help of JDBC & SQL. Create MS Access Database, Create on ODBC link, Compile & execute JAVA JDVC Socket.
9. Design and implement a simple shopping cart example with session tracking API.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB II	MCA 208	DATABASE MANAGEMENT SYSTEM LAB	L	T	P	C
			0	0	3	2
GOAL	To design and develop database applications					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Learn to write a query 2. Learn to design a simple dB using data modelling techniques 3. To provide the knowledge of various dB tools			The student should be able to 1. Populate and query a database using SQL DML/DDDL commands 2. Write programs using PL/SQL including stored procedures , cursors, packages etc. 3. Construct real time database application using current techniques			

The programme to be implemented using SQL

1. Create Table, SQL for Insertion, Deletion, Update and Retrieval using aggregating functions.
2. Write Programs in PL/SQL, Understanding the concept of Cursors.
3. Write Program for Join, Union & intersection etc.
4. Creating Views, Writing Assertions, Triggers.
5. Creating Forms, Reports etc.
6. Writing codes for generating read and update operator in a transaction using different situations.
7. Implement of 2PL concerning central algorithm.
8. Developing code for understanding of distributed transaction processing.

Students are advised to use Developer 2000 Oracle 8+ version for above experiments. However, depending on the availability of Software's students may use power builder/SQL Server/DB2 etc. for implementation.

SQL COMMANDS]

1. Using Select queries , where clause, order by clause, distinct keywords etc on a sample database. Create a sample Relational Database.
- 2) To create a table, alter and drop table.
- 3) To perform select, update, insert and delete operation in a table.
- 4) To make use of different clauses viz where, group by, having, order by, union, interUNIT, set difference.
- 5) To study different constraints.

[SQL FUNCTION]

- 6) To use oracle function viz aggregate, numeric, conversion, string function.
- 7) To understand use and working with joins.
- 8) To understand use and working of sub-queries.
- 9) To make use of transaction control statement viz rollback, commit and save point.
- 10) To make views of a table.
- 11) To make indexes of a table.
- 12) To inbuilt SQL function to create database.

[PL/SQL]

- 13) Introduction to SQL & PL/SQL

- 14) To implement Cursor on a table.
- 15) To implement trigger on a table
- 16) Creating Procedures and Function.
- 17) To implement control structure.
- 18) To implement Packages.

LIST OF EXPERIMENTS

1. To study Basic SQL commands (create database, create table, use , drop, insert) and execute the following queries using these commands:

- ☐ Create a database named 'Employee'.
- ☐ Use the database 'Employee' and create a table 'Emp' with attributes 'ename', 'ecity', 'salary', 'enumber', 'eaddress', 'deptname'.
- ☐ Create another table 'Company' with attributes 'cname', 'ccity', 'empnumber' in the database 'Employee'.

2. To study the viewing commands (select , update) and execute the following queries using these commands:

- ☐ Find the names of all employees who live in Delhi.
- ☐ Increase the salary of all employees by Rs. 5,000.
- ☐ Find the company names where the number of employees is greater than 10,000.
- ☐ Change the Company City to Gurgaon where the Company name is 'TCS'.

3. To study the commands to modify the structure of table (alter, delete) and execute the following queries using these commands:

- ☐ Add an attribute named 'Designation' to the table 'Emp'.
- ☐ Modify the table 'Emp', Change the datatype of 'salary' attribute to float.
- ☐ Drop the attribute 'deptname' from the table 'emp'.
- ☐ Delete the entries from the table 'Company' where the number of employees are less than 500.

4. To study the commands that involve compound conditions (and, or, in , not in, between , not between , like , not like) and execute the following queries using these commands:

- ☐ Find the names of all employees who live in 'Gurgaon' and whose salary is between Rs. 20,000 and Rs. 30,000.
- ☐ Find the names of all employees whose names begin with either letter 'A' or 'B'.

- Find the company names where the company city is 'Delhi' and the number of employees is not between 5000 and 10,000.

- ☐ Find the names of all companies that do not end with letter 'A'.

5. To study the aggregate functions (sum, count, max, min, average) and execute the following queries using these commands:

- ☐ Find the sum and average of salaries of all employees in computer science department.
- ☐ Find the number of all employees who live in Delhi.
- ☐ Find the maximum and the minimum salary in the HR department.

6. To study the grouping commands (group by, order by) and execute the following queries using these commands:

- ☐ List all employee names in descending order.

- ☐ Find number of employees in each department where number of employees is greater than 5.
 - ☐ List all the department names where average salary of a department is Rs.10,000.
7. To study the commands involving data constraints and execute the following queries using these commands:
- ☐ Alter table 'Emp' and make 'enumber' as the primary key.
 - ☐ Alter table 'Company' and add the foreign key constraint.
 - ☐ Add a check constraint in the table 'Emp' such that salary has the value between 0 and Rs.1,00,000.
 - ☐ Alter table 'Company' and add unique constraint to column cname.
 - ☐ Add a default constraint to column ccity of table company with the value 'Delhi'.
8. To study the commands for aliasing and renaming and execute the following queries using these commands:
- ☐ Rename the name of database to 'Employee1'.
 - ☐ Rename the name of table 'Emp' to 'Emp1'.
 - ☐ Change the name of the attribute 'ename' to 'empname'.
9. To study the commands for joins (cross join, inner join, outer join) and execute the following queries using these commands:
- ☐ Retrieve the complete record of an employee and its company from both the table using joins.
 - ☐ List all the employees working in the company 'TCS'.
10. To study the various set operations and execute the following queries using these commands:
- ☐ List the enumber of all employees who live in Delhi and whose company is in Gurgaon or if both conditions are true.
 - ☐ List the enumber of all employees who live in Delhi but whose company is not in Gurgaon.
11. To study the various scalar functions and string functions (power, square, substring, reverse, upper, lower, concatenation) and execute the following queries using these commands:
- ☐ Reverse the names of all employees.
 - ☐ Change the names of company cities to uppercase.
 - ☐ Concatenate name and city of the employee.
12. To study the commands for views and execute the following queries using these commands:
- ☐ Create a view having ename and ecity.
 - ☐ In the above view change the ecity to 'Delhi' where ename is 'John'.
 - ☐ Create a view having attributes from both the tables.
 - ☐ Update the above view and increase the salary of all employees of IT department by Rs.1000.
13. To study the commands involving indexes and execute the following queries:
- ☐ Create an index with attribute ename on the table employee.
 - ☐ Create a composite index with attributes cname and ccity on table company.
 - ☐ Drop all indexes created on table company.
14. To study the conditional controls and case statement in PL-SQL and execute the following queries:
- ☐ Calculate the average salary from table 'Emp' and print increase the salary if the average salary is less than 10,000.
 - ☐ Print the deptno from the employee table using the case statement if the deptname is 'Technical' then deptno is 1, if the deptname is 'HR' then the deptno is 2 else deptno is 3.

15. To study procedures and triggers in PL-SQL and execute the following queries:

- ☐ Create a procedure on table employee to display the details of employee to display the details of employees by providing them value of salaries during execution.
- ☐ Create a trigger on table company for deletion where the whole table is displayed when delete operation is performed.

16. 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)

a. Create the above tables by properly specifying the primary keys and foreign keys.

b. Enter at least five tuples for each relation.

c. Demonstrate how you

➤ Update the damage amount for the car with specific regno in the accident with report number 12 to 25000.

➤ Add a new accident to the database.

d. Find the total number of people who owned cars that were involved in accidents in the year 2008.

e. Find the number of accidents in which cars belonging to a specific model were involved.

17. Consider the following relations for a 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)

a. Create the above tables by properly specifying the primary keys and foreign keys.

b. Enter at least five tuples for each relation.

c. 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.

d. List the orderno for orders that were shipped from all the warehouses that the company has in a specific city.

e. 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.

18. 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)

a. Create the above tables by properly specifying the primary keys and foreign keys.

b. Enter atleast five tuples for each relation.

c. Demonstrate how you add a new text book to the database and make this book to be adopted by some department.

d. 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.

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

19. 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)

- a. Create the above tables by properly specifying the primary keys and foreign keys.
- b. Enter at least five tuples for each relation.
- c. 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.
- d. Find the author of the book that has maximum sales.
- e. Demonstrate how you increase the price of books published by a specific publisher by 10%.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB III	MCA 209	ADVANCED JAVA PROGRAMMING LAB	L	T	P	C
			0	0	3	2
GOAL	To write and execute programs in JAVA					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Practice logical ability to solve the problems. 2. Understand java programming development environment, compiling, debugging, linking and executing a program using the development environment 3. Understand and apply the in-built functions and customized functions for solving the problems. 4. Study, analyze and understand logical structure of a computer program, and different construct to develop a program in Java Programming language			The student should be able to 1. Apply decision and iteration control structures to implement algorithms in Java 2. Able to implement String and string buffer methods 3. Implement Complex number operations 4. Implement inheritance, polymorphism and object relationship in java 5. Implement interfaces as programming techniques 6. Able to implement Packages 7. Analyze and create Applet Programs 8. Apply exceptions handling 9. Able to generate multiple threads			

LIST OF EXPERIMENTS

1. Write a program to print the individual digits of a 3-digit number.
2. Write a program that asks the user to enter two integers, obtains the numbers from the user, and then prints the larger number followed by the words "is larger." If the numbers are equal, print the message —These numbers are equal.¶
3. Write a program to read N numbers and find the largest and smallest numbers.
4. Write a program to create a String object. Initialize this object with your name. Find the length of your name using the appropriate String method. Find whether the character ‘a’ is in your name or not; if yes find the number of times ‘a’ appears in your name. Print locations of occurrences of ‘a’.
5. Write a program to create a StringBuffer object and illustrate the following
 - ☐ Display the capacity and length of the string buffer.
 - ☐ insert characters at the beginning.
 - ☐ the append() and reverse() functions
6. Write a program to display total marks of 5 students using student class. Given the following attributes: Regno(int), Name(string), Marks in subjects(Integer Array), Total (int).
7. Write a program in Java with class Rectangle with the data fields width, length, area and colour. The length, width and area are of double type and colour is of string type. The methods are get_length(), get_width(), get_colour() and find_area().
Create two objects of Rectangle and compare their area and colour. If the area and colour both are the same for the objects then display — Matching Rectangles¶, otherwise display — Non-matching Rectangle¶.
8. Write a program to create a player class. Inherit the classes Cricket_player, Football_player and Hockey_player from player class.
9. Write a program to show how a class implements two interfaces.
10. Show through a program that fields in an interface are implicitly static and final and methods are automatically public.
11. Write a program to create a package for Book details giving Book Name, Author Name, Price, year of publishing.
12. A color can be created by specifying the red, green, blue values as integer parameters to the constructor of class Color. The values range from 0 to 255. Provide three horizontal scroll bars and ask

the user to select the values of the colors by dragging the thumb in the scroll bar. Using the color selected, draw a rectangle.

13. Write a java program to copy the contents of one file to another file.
14. Write a Java program to read input from the standard input and write to a byte array.
15. Create an applet for simple calculator to perform Addition, Subtraction, Multiplication and Division using Button, label and Text field classes.
16. Write a Java program to catch more than two exceptions.
17. Write a Java program to create your own exception subclass that throws exception if the sum of two integers is greater than 99.
18. Write a Java program for generating two threads, one for printing even umbers and other for printing odd numbers.
19. Write a Java program for producer and consumer problem using Thread.
20. Write a program in Java for illustrating, overloading, over riding and various forms of inheritance.
21. Write programs to create packages and multiple threads in Java.
22. Write programs in Java for event handling Mouse and Keyboard events.
23. Using Layout Manager create different applications.
24. Write programs in Java to create and manipulate Text Area, Canvas, Scroll Bars, Frames and Menus using swing/AWT.
25. Using Java create Applets.
25. Use Java Language for Client Server Interaction with stream socket connections.
26. Write a program in java to read data from disk file.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB IV	MCA 210	DESIGN & ANALYSIS OF ALGORITHMS LAB	L	T	P	C
			0	0	3	2
GOAL	To implement different algorithmic techniques and analyze an efficiency of algorithms.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Realize the concept of Divide and Conquer algorithm design technique 2. Understand the concept of Dynamic Programming algorithm design technique 3. Realize the concept of Greedy algorithm design technique 4. Understand the concept of Backtracking algorithm design technique			The student should be able to 1. Implement Divide and Conquer algorithm design technique for various applications 2. Implement dynamic programming algorithm design technique for various applications 3. Implement Greedy algorithm design technique for various applications 4. Implement backtracking algorithm design technique for various applications 5. Ability to write programs in java to solve problems using algorithm design techniques such as Divide and Conquer, Greedy, Dynamic programming, and Backtracking.			

LIST OF EXPERIMENTS

1. Write a program in C to implement Binary Search using Divide and Conquer Method.
2. Write a program in C to implement MaxMin Problem using Divide and Conquer Method
3. Write a program in C to implement mergesort using Divide and Conquer Method
4. Write a program in C to implement all pairs shortest path using dynamic programming
5. Write a program in C to implement travelling salesman problem using dynamic programming
6. Write a program in C to solve Knapsack Problem using greedy techniques
8. Write a program in C to solve 8-Queens Problem using Back tracking
9. Write a C program to implement Quick sort algorithm for sorting a list of integers in ascending order
10. Write a C program to implement Heap sort algorithm for sorting a list of integers in ascending order
11. Write a C program to implement Insertion sort algorithm for sorting a list of integers in ascending order
12. Write a C program to implement bubble sort algorithm for sorting a list of integers in ascending order
13. Write a C program to implement selection sort algorithm for sorting a list of integers in ascending order
14. Write a C program to implement Merge sort algorithm for sorting a list of integers in ascending order.
15. Write a C program to implement the dfs algorithm for a graph.
16. Write a C program to implement the bfs algorithm for a graph.
17. Write a C programs to implement backtracking algorithm for the N-queens problem.
18. Write a C program to implement the backtracking algorithm for the sum of subsets problem.
19. Write a C program to implement the backtracking algorithm for the Hamiltonian Circuits problem.
20. Write a C program to implement greedy algorithm for job sequencing with deadlines.
21. Write a C program to implement Dijkstra's algorithm for the Single source shortest path problem.

22. Write a C program that implements Prim's algorithm to generate minimum cost spanning tree.
23. Write a C program that implements Kruskal's algorithm to generate minimum cost spanning tree
24. Write a C program to implement Floyd's algorithm for the all pairs shortest path problem.
25. Write a C program to implement Dynamic Programming algorithm for the 0/1 Knapsack problem.
26. Write a C program to implement Dynamic Programming algorithm for the Optimal Binary Search Tree Problem.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB V	MCA 211	SOFTWARE ENGINEERING LAB	L	T	P	C
			0	0	3	2
GOAL	To explore various techniques in modern software engineering					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Understand, design and develop software and also perform testing through test cases 2. To understand the software engineering methodologies involved in the phases for project development. 3. To gain knowledge about open source tools used for implementing software engineering methods. 4. To exercise developing product-startups implementing software engineering methods. 5. Open source Tools: StarUML / UMLGraph / Topcased			The student should be able to 1. Develop a model for Requirement. 2. Design Software Engineering projects using UML 3. Use testing tools to do software testing. 4. Use version control tools and create build files			

LIST OF EXPERIMENTS

Every student must be given one unique business problem/customer requirement to work on the practical using modern software engineering.

- Application of traditional software engineering process for a case study to develop a model for the business requirement
- Application of SCRUM for a case study to develop a software model
- Application of ADP for a case study to develop a software model
- Come up with system design – UI & Database Design
- Develop User defined types, exception handlers and concurrency control mechanisms
- Implement the code / software product based on the requirement
- Debugging and Testing of the code using manual and automated testing tools
- SCCS and MAKE File to create builds
- Preparation of Project Documentation and Training Manual

Prepare the following documents and develop the software project startup, prototype model, using software engineering methodology for at least two real time scenarios or for the sample experiments.

Problem Analysis and Project Planning -Thorough study of the problem – Identify Project scope, Objectives and Infrastructure.

Software Requirement Analysis – Describe the individual Phases/modules of the project and Identify deliverables. Identify functional and non-functional requirements.

Data Modeling – Use work products – data dictionary.

Software Designing – Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.

Prototype model – Develop the prototype of the product.

Course management system (CMS)

A course management system (CMS) is a collection of software tools providing an online environment for course interactions. A CMS typically includes a variety of online tools and environments, such as:

- An area for faculty posting of class materials such as course syllabus and handouts

- An area for student posting of papers and other assignments
- A grade book where faculty can record grades and each student can view his or her grades
- An integrated email tool allowing participants to send announcement email messages to the entire class or to a subset of the entire class
- A chat tool allowing synchronous communication among class participants
- A threaded discussion board allowing asynchronous communication among participants

In addition, a CMS is typically integrated with other databases in the university so that students enrolled in a particular course are automatically registered in the CMS as participants in that course.

The Course Management System (CMS) is a web application for department personnel, Academic Senate, and Registrar staff to view, enter, and manage course information formerly submitted via paper. Departments can use CMS to create new course proposals, submit changes for existing courses, and track the progress of proposals as they move through the stages of online approval.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB VI	MCA 212	IOT PROJECT / SEMINAR*	L	T	P	C
			0	0	3	2
GOAL	1. The project goal refers to achieving a desired outcome (Theory and Practical knowledge of IOT) at a specific end date employing a specific amount of resources. 2. Seminar may be for the purpose of education, such as a lecture, where the participants engage in the discussion of an academic subject for the aim of gaining a better insight into the subject. 3. The purpose of this course is to impart knowledge on Internet of Things (IoT), which relates to the study of sensors, actuators, and controllers, among other Things, IoT applications and examples overview (building automation, transportation, healthcare, industry, etc.) with a focus on wearable electronics					
OBJECTIVES			OUTCOMES			
The course should enable the students to: The students will learn: a) IOT concepts b) IOT Standards c) Components of IOT System. d) Relevance of IOT for the future. e) IOT Applications. f) IOT for smart cities (Case study Smart city Barcelona) g) IOT in Indian Scenario h) Challenges in IOT implementation.			At the end of the course the student should be able to: 1. To develop of IoT value chain structure (device, data cloud), application areas and technologies involved 2. To develop IoT sensors and technological challenges faced by IoT devices, with a focus on wireless, energy, power, RF and sensing modules 3. To develop Market forecast for IoT devices with a focus on sensors 4. To develop Explore and learn about Internet of Things with the help of preparing projects designed for Raspberry Pi			

*Seminar means giving explanation for a group of people. Paper presentation means presenting an explanation on a specified concept may include seminar in it.

*Project means doing for result oriented it may include both above two.

Students make a project thesis in specific IOT topic. Maximum Two Students may be participated in one Project. After developing Project thesis student presenting an explanation on a specified concept may include seminar in it. Students develop project under the guidance of faculty members.

UNIT I:

Introduction: Internet of Things Promises , Definition , Scope , Sensors for IoT Applications , Structure of IoT , IoT Map Device

UNIT II:

Seven Generations of IOT Sensors To Appear :Industrial sensors, Description & Characteristics , First Generation, Description & Characteristics , Advanced Generation, Description & Characteristics , Integrated IoT Sensors, Description & Characteristics , Polytronics Systems, Description & Characteristics , Sensors' Swarm, Description & Characteristics , Printed Electronics, Description & Characteristics , IoT Generation Roadmap

UNIT III:

Technological Analysis: Wireless Sensor Structure , Energy Storage Module , Power Management Module , RF Module , Sensing Module

UNIT IV:

IOT Development Examples: ACOEM Eagle, EnOcean Push Button, NEST Sensor, Ninja Blocks, Focus on Wearable Electronics

UNIT V:

Preparing IOT Projects: Creating the sensor project, Preparing Raspberry Pi, Clayster libraries, HardwareInteracting with the hardware, Interfacing the hardware , Internal representation of sensor values, Persisting data, External representation of sensor values, Exporting sensor data, Creating the actuator project , Hardware, Interfacing the hardware, Creating a controller, Representing sensor values, Parsing sensor data, Calculating control states, Creating a camera, Hardware, Accessing the serial port on Raspberry Pi, Interfacing the hardware, Creating persistent default settings, Adding configurable properties, Persisting the settings, Working with the current settings, Initializing the camera

References

1. Dr. Guillaume Girardin , Antoine Bonnabel, Dr. Eric Mounier, 'Technologies & Sensors for the Internet of Things Businesses & Market Trends 2014, 2024', Yole Développement Copyrights ,2014
2. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015
3. Editors OvidiuVermesan Peter Friess,'Internet of Things, From Research and Innovation to Market
4. Deployment', River Publishers, 2014
5. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.
6. The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World http://www.amazon.in/Internet-Things-Smart-CitiesChanging/dp/0789754002/ref=sr_1_9?ie=UTF8&qid=1474003280&sr=8-9&keywords=internet+of+things+book ...

Prepare Project Thesis according Appendix 1

**DR. RAMMANOHAR LOHIA AVADH UNIVERSITY
AYODHYA**



Evaluation Scheme & Syllabus

for

MCA –DEGREE

On

Choice Based Credit System

(Effective from the Session: 2020-21)

Dr. RamManohar Lohia Avadh University, Ayodhya U.P.

Study and Evaluation Scheme

MCA (Master of Computer Applications)

(Effective from Session 2021-22)

Year – II Semester – III

Sl. No.	Subject Code	Subject Name	Periods			Evaluation Scheme					Credit
			L	T	P	Sessional Exams			ESE	Subject Total	
						CT	TA	Total			
	THEORY SUBJECT										
1	MCA 301	Android Programming & Mobile Computing	3	1	0	30	20	50	100	150	04
2	MCA 302	PHP Programming	3	1	0	30	20	50	100	150	04
3	MCA 303	Computer Network	3	1	0	30	20	50	100	150	04
4	MCA 304	Dot Net Framework & C#	3	1	0	30	20	50	100	150	04
5	MCA 305	Artificial Intelligence	3	1	0	30	20	50	100	150	04
6	MCA 306	Elective – II	3	1	0	30	20	50	100	150	04
	Practical										
7	MCA 307	Android Programming Lab	0	0	3	10	10	20	30	50	02
8	MCA 308	PHP Programming Lab	0	0	3	10	10	20	30	50	02
9	MCA 309	Computer Network Lab	0	0	3	10	10	20	30	50	02
10	MCA 310	Dot Net Framework & C# Lab	0	0	3	10	10	20	30	50	02
11	MCA 311	Artificial Intelligence Lab	0	0	3	10	10	20	30	50	02
12	MCA 312	Industrial Project Viva *	0	0	3	10	10	20	30	50	02
		Total	18	6	18	-	-	-	-	1200	36

Elective – II

MCA 306-(i)	Automata and Compiler design
MCA 306-(ii)	Database Security
MCA 306-(iii)	Cryptography and Network Security
MCA 306-(iv)	Software and Application Security
MCA 306-(v)	Neural Network and Fuzzy Logic
MCA 306-(vi)	Wireless Sensor Networks

CT: Class Test

TA: Teacher Assessment

L/T/P: Lecture/Tutorial/ Practical

MSE: Mid Semester Examination

ESE: End Semester Examination

*End of MCA 2nd Semester Exam students will do 45 days Industry Internship/ training which is based on Latest technology.

Students make a project thesis based on Industry Internship / training which you done after IIND Sem End Semester Examination. Maximum Two Students may be participated in one Project. After developing Project thesis student presenting an explanation on a specified concept may include seminar in it. Students develop project under the guidance of faculty members / Industry Expert. Student Submitted Industry Internship/ training Certificate with Project Thesis.

Prepare Project Thesis according MCA Thesis Guidelines 2020

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER I	MCA 301	ANDROID PROGRAMMING & MOBILE COMPUTING	L	T	P	C
			3	1	3	4
GOAL	To impart knowledge on various aspects of Android programming and mobile computing					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To have a review on concept of Android programming. 2. To learn Android Programming Environments. 3. To practice programming in Android. 4. To learn GUI Application development in Android platform with XML 5. Learn mobile computing architecture. 6. Learn various mobile technologies. 3. Learn GPRS and WAP technologies. 7. Learn Wireless LAN technology. 8. Learn platforms for Mobile Application development..			The student should be able to 1. Understand the various Android programming architectures. 2. Design and develop mobile apps, using Android as development platform, with key focus on user experience design, native data handling and background tasks and notifications. 3. Understand the various mobile architectures. 4. Understand various messaging environment in mobile environments. 5. Acquire knowledge to develop mobile applications under PALM and SYMBIAN OSs. 6. Write mobile applications using J2ME.			

UNIT I:

Android Programming: Introducing the android computing platform, History of android, android software stack, Developing end user application using android SDK, android java packages, Setting up the development environment, Installing android development tools (ADT), Fundamental components, Android virtual devices, Running on real device, Structure of android application, Application life cycle.

Understanding android resources - String resources, Layout resources, Resource reference syntax, Defining own resource IDs - Enumerating key android resources, string arrays, plurals, Colour resources, dimension resources, image resources, Understanding content providers - android built in providers, exploring databases on emulator, architecture of content providers, structure of android content URIs, reading data using URIs, using android cursor, working with where clause, inserting updates and deletes, implementing content, Understanding intents – basics of intents, available intents, exploring intent composition, Rules for Resolving Intents to Their Components, ACTION PICK, GET CONTENT, pending intents

UNIT II:

User interfaces development in android - building UI completely in code, UI using XML, UI in XML with code, Android's common controls - Text controls, button controls, checkbox control, radio button controls, image view, date and time controls, map view control, understanding adapters, adapter views, list view, grid view, spinner control, gallery control, styles and themes, Understanding layout managers - linear layout manager, table layout manager, relative layout manager, frame layout manager, grid layout manager.

UNIT III:

Android menus - creating menus, working with menu groups, responding to menu items, icon menu, sub menu, context menu, dynamic menus, loading menu through XML, popup menus, Fragments in android - structure of fragment, fragment life cycle, fragment transaction and back stack, fragment manager, saving fragment state, persistence of fragments, communications with fragments, startActivity() and set- TargetFragment(), using dialogs in android, dialog fragments, working with toast, Implementing action bar - tabbed navigation action bar activity, implementing base activity classes, tabbed action bar and tabbed listener, debug text view layout, action bar and menu interaction,

Isit navigation action bar activity, spinner adapter, list listener, list action bar, standard navigation action bar activity, action bar and search view, action bar and fragments.

Persisting data - Files, saving state and preferences - saving application data, creating, saving and retrieving shared preferences, preference framework and preference activity, preference layout in XML, native preference controls, preference fragments, preference activity, persisting the application state, including static files as resources, Working with file system, SQLite - SQLite types, database manipulation using SQLite, SQL and database centric data model for android, android database classes.

UNIT IV:

Introduction: Mobility of Bits and Bytes, Wireless, The Beginning, Mobile Computing, Dialogue Control, Networks, Middleware and Gateways, Application and Services (Contents), Developing Mobile Computing Applications, Security in Mobile Computing, Standards, Why is it Necessary?, Standard Bodies, Players in the Wireless Space.

Mobile Computing Architecture: Internet, The Ubiquitous Network, Architecture for Mobile Computing, Three-Tier Architecture, Design Considerations for Mobile Computing, Mobile Computing through Internet, Making Existing Applications Mobile, Enabled.

Mobile Computing Through Telephony: Evolution of Telephony, Multiple Access Procedures, Mobile Computing through Telephone, Developing an IVR Application, Voice XML, Telephony Application Programming Interface (TAPI).

Emerging Technologies: Introduction, Bluetooth, Radio Frequency Identification (RFID), WiMAX, Mobile IP, IPv6, Java Card.

Global System for Mobile Communications (GSM): GSM Architecture, Entities, Call Routing in GSM, PLMN Interfaces, GSM Addresses and Identifiers, Network Aspects in GSM, GSM Frequency Allocation, Authentication and Security.

SMS, GPRS AND WAP: Short Message Service (SMS): Mobile Computing over SMS, SMS, Value Added Services through SMS, Accessing the SMS Bearer.

GPRS: Packet Data Network, Network Architecture, Network Operations, Data Services in GPRS, Applications for GPRS, Limitations, Billing and Charging.

Wireless Application Protocol (WAP): Introduction, WAP, MMS, GPRS Applications.

UNIT V:

CDMA and 3G: Introduction, Spread Spectrum Technology, IS-95, CDMA Vs GSM, Wireless Data, 3G Networks & Applications.

Wireless LAN: Introduction, Advantages, IEEE 802.11 Standards, Architecture, Mobility, Deploying, Mobile Ad Hoc Networks and Sensor Networks, Wireless LAN Security, WiFi Vs 3G.

Internet Networks and Interworking: Introduction, Fundamentals of Call Processing, Intelligence in the Networks, SS#7 Signaling, IN Conceptual Model, Softswitch, Programmable Networks, Technologies and Interfaces for IN.

Client Programming: Introduction, Moving Beyond the Desktop, A Peek under the Hood: Hardware Overview, Mobile Phones, PDA, Design Constraints in Applications for Handheld Devices.

Wireless Devices with SYMBIAN OS: Introduction, Symbian OS Architecture, Applications for Symbian, Control and Compound Controls, Active Objects, Localization, Security on the Symbian OS.

Programming for the Android OS: Introduction, Android Architecture, Application Development. **J2ME:** JAVA in the Handset, Three-Prong Approach to Java Everywhere, Java 2 Micro Edition (J2ME), Programming for CLDC, GUI in MIDP, UI Design Issues, Multimedia, Record Management System, Communication in MIDP, Security Considerations in MIDP, Optional Packages

Text Book

1. Asoke K Talukder & Roopa R. Yavagal, —Mobile Computing, Technology Applications and Service Creation, TMH 2006.

Reference

1. Uwe Hansmann, Lothar Merk, Martin S.Nicklous, Thomas Staber, —Principles of ComputingII, 2/e, Springer International Edition.
1. J. Schiller, “Mobile Communications”, Addison Wesley.
2. Charles Perkins, “Mobile IP”, Addison Wesley.
3. Charles Perkins, “Ad hoc Networks”, Addison Wesley.
4. Upadhyaya, “Mobile Computing”, Springer New York.
5. *Pro Android 4*, Satya Komatineni & Dave MacLean, Apress.
6. *Professional Android 4 Application Development*, Retomeier, Wrox.
7. *Programming Android*, Zigurd

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER II	MCA 302	PHP PROGRAMMING	L	T	P	C
			3	1	3	4
GOAL	The Objective is to familiarize students with the latest trends and advancements in PHP programming Languages which are used in industries					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Familiarize students with the coding of PHP 2. To practice PHP Programming. 3. To practice developing dynamic websites. 4. To practice how to interact with databases through PHP.			The student should be able to 1. Apply the concepts studies in PHP coding languages 2. Code in PHP and Javascript 3. To develop Static websites or Dynamic websites or Web applications 4. Write PHP scripts to handle HTML forms. 5. Write regular expressions including modifiers, operators, and metacharacters. Create PHP programs that use various PHP library functions, and that manipulate files and directories. 6. Analyze and solve various database tasks using the PHP language. 7. Analyze and solve common Web application tasks by writing PHP programs.			

UNIT I:

PHP - Introduction, Environment Setup, Syntax Overview, Variable Types, Constants, Operator Types, Decision Making, Loop Types, Arrays, Strings, Web Concepts, GET & POST,

UNIT II:

Advanced PHP: File Inclusion, Files & I O, Functions, Cookies, Sessions, Sending Emails, File Uploading, Predefined Variables, Regular Expression, Error Handling, Bugs Debugging, Date & Time, PHP & MySQL, PHP & AJAX, PHP & XML, Object Oriented, PHP - For C Developers, PHP - For PERL Developers

UNIT III:

PHP Form Examples: Form Introduction, Validation Example, Complete Form

PHP login Examples: Login Example, Facebook Login, Paypal Integration, MySQL Login,

UNIT IV:

PHP & MySQL: Features of MySQL, data types, Introduction to SQL commands - SELECT, DELETE, UPDATE, INSERT, PHP functions for MySQL operations: mysql_connect, mysql_select_db, mysql_query, mysql_fetch_row, mysql_fetch_array, mysql_fetch_object, mysql_result, Insertion and Deletion of data using PHP, Displaying data from MYSQL in webpage.

PHP AJAX Examples: Introduction to AJAX, Implementation of AJAX in PHP, Simple examples like partial page update, Concept of master page, applying templates, AJAX Search, AJAX XML Parser, AJAX Auto Complete Search, AJAX RSS Feed Example

UNIT V:

PHP XML Example: PHP - XML Introduction, Simple XML, SAX Parser Example, DOM Parser Example

PHP Frame Works: Frame Works, Core PHP vs Frame Works

PHP Design Patterns: Design Patterns, PHP Function Reference, PHP - Built-In Functions

References

1. The Joy of PHP Programming: A Beginner's Guide – by Alan Forbes. ...
2. PHP & MySQL Novice to Ninja – by Kevin Yank. ...
3. Head First PHP & MySQL – by Lynn Beighley & Michael Morrison. ...
4. Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites – by Robin Nixon.
5. PHP & MySQL Web Development – by Luke Welling & Laura Thompson
6. PHP & MySQL: The Missing Manual – by Brett McLaughlin
7. PHP: A Beginner's Guide – by Vikram Vaswani
8. Learn PHP & MySQL – Zero to Hero Programming Crash Course – by Paul Madoff
9. Murach's PHP & MySQL – by Joel Murach & Ray Harris
10. Programming PHP – by Kevin Tatroe, Peter MacIntyre & Rasmus Lerdorf “Foreword By: Michael Bourque”

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER III	MCA 303	COMPUTER NETWORK	L	T	P	C
			3	1	3	4
GOAL	To introduce the concepts, terminologies and technologies used in modern data communication and computer networks.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Understand the concepts of data communications. 2. Study the functions of the data link layer and to introduce IEEE standards employed in computer networking. 3. Learn the functions of the network layer and to get familiarized with the different protocols involved. 4. Learn the functions of the transport layer and to get familiarized with the different protocols involved. 5. Understand multiplexing , Domain name space and protocols.			The student should be able to 1. Describe various components and categories of data communications, types of connections, topologies, protocols and standards, various transmission media and modems. 2. Detect and correct the errors using various algorithmic techniques, be aware of the various Ethernet standards and bridges. 3. Explain various switching techniques used and implement the various routing and router protocols. 4. Illustrate multiplexing and demultiplexing, UDP, TCP protocols and Congestion Control mechanisms. 5. Illustrate Network Applications.			

UNIT I:

Introduction Concepts: Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design - Delay Analysis, Back Bone Design, Local Access Network Design, Physical Layer Transmission Media, Switching methods, ISDN, Terminal Handling.

UNIT II:

Medium Access sub layer: Medium Access sub layer - Channel Allocations, LAN protocols - ALOHA protocols - Overview of IEEE standards - FDDI. Data Link Layer - Elementary Data Link Protocols, Sliding Window protocols, Error Handling. Internetworks , Packet Switching and Datagram approach , IP addressing methods , Subnetting , Routing, Distance Vector Routing, Link State Routing , Routers.

UNIT III:

Network Layer: Network Layer - Point - to Pont Networks, routing, TCP / IP, IP packet, IP address, IPv6.

Network Application: IP Camera Management: IP Camera Technical specifications, IP Camera form factor, IP Camera Image features, D Link, IP Camera Advanced Configurations, Switch Management: Overview of D Link Switches and features, Accessing and configuration the switch, Switch learning process, VLAN and GVRP, Understanding Spanning Tree protocol, Switch life cycle, Basic-Understanding D Link Switching features like DHCP, ACL, LDP and System Maintenance

UNIT IV:

Transport Layer: Transport Layer - Design issues, connection management, session Layer-Design issues, remote procedure call. Presentation Layer-Design issues, Data compression techniques, cryptography - TCP - Window Management, Duties of transport layer, Multiplexing, Demultiplexing , Sockets , User Datagram Protocol (UDP) , Transmission Control Protocol (TCP) , Types of TCP

Unit-V

Application Layer: Application Layer: File Transfer, Access and Management, Electronic mail, Virtual Terminals, Other application. Example Networks - Internet and Public Networks. Congestion Control , Quality of services (QOS), Integrated Services, Domain Name Space (DNS), Flux, Fast Flux , SMTP, FTP , HTTP, WWW , Security, Cryptography.

Text Books

1. Forouzen, "Data Communication and Networking", TMH
2. A.S. Tanenbaum, Computer Networks, Pearson Education
3. W. Stallings, Data and Computer Communication, Macmillan Press

References

1. Anuranjan Misra, "Computer Networks", Acme Learning
 2. G. Shanmugarathinam, "Essential of TCP/ IP", Firewall Media
 3. James F. Kurose and Keith W. Ross, —Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Ed., Pearson Education, 2003.
 4. L.Peterson and Peter S. Davie, —Computer Networks, 5th Ed., Morgan Kaufmann, 2011.
 5. Andrew S. Tanenbaum, —Computer Networks, 5th Ed., Prentice Hall, 2010.
- William Stallings, —Data and Computer Communication, 8th Ed., Pearson, 2006.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER IV	MCA 304	DOT NET FRAMEWORK & C#	L	T	P	C
			3	1	3	4
GOAL	To learn the fundamental concepts in .NET framework and programming.					
OBJECTIVES		OUTCOMES				
The course should enable the students to 1. Gain knowledge in the concepts of the .NET framework as a whole and the technologies that constitute the framework. 2. Write Programs in C# and ASP.Net, both in basic and advanced levels. 3. Build sample applications. 4. Develop an appreciation of key generic .Net concepts and techniques.		The student should be able to 1. Learn the basis of .Net framework. 2. Understand object oriented Aspects of C# and ASP.Net. 3. Develop simple applications under .Net framework. 4. Develop Web based Applications using .Net programming languages.				

UNIT I:

The .Net framework: Introduction, The Origin of .Net Technology, Common Language Runtime (CLR), Common Type System (CTS), Common Language Specification (CLS), Microsoft Intermediate Language (MSIL), Just-In –Time Compilation, Framework Base Classes.

UNIT II:

C -Sharp Language (C#): Introduction, Data Types, Identifiers, Variables, Constants, Literals, Array and Strings, Object and Classes, Inheritance and Polymorphism, Operator Overloading, Interfaces, Delegates and Events. Type conversion.

UNIT III:

C# Using Libraries: Namespace- System, Input-Output, Multi-Threading, Networking and sockets, Managing Console I/O Operations, Windows Forms, Error Handling, Work with the Windows Forms and controls, Perform validation of controls using classes and controls , Work with Dialog Boxes, Menus and MDI Application, Implement Printing and Reporting Functionality in a Windows Forms Application, Package and deploy applications.

UNIT IV:

Advanced Features Using C#: Web Services, Window Services, Asp.net Web Form Controls, ADO.Net. Create and manage connections using ADO.NET, Identify the disconnected and connected environment in ADO.NET, Create datasets and data tables, Retrieve and store large binary data, Perform bulk copy operations , Execute SQL notification maintain and update a cache, Read, write, validate, and modify XML data using XML reader and writer classes. Distributed Application in C#, Unsafe Mode, Graphical Device interface with C#.

UNIT V:

.Net Assemblies and Attribute: .Net Assemblies features and structure, private and share assemblies, Built-In attribute and custom attribute. Introduction about generic.

Developing Web Applications Using Asp.Net: Create a Web Application, Program a Web Application, Add and Configure Server Controls, Create a Common Layout by Using Master Pages, Manage State for a Web Application , Access and Display Data, Control Access to a Web Application, Deploy a Web Application, Build Dynamic Web Applications, Create Controls for Web Applications.

References

1. Wiley," Beginning Visual C# 2008",Wrox
2. Fergal Grimes," Microsoft .Net for Programmers". (SPI)

3. Balagurusamy,” Programming with C#”, (TMH)
4. Mark Michaelis, “Essential C# 3.0: For .NET Framework 3.5, 2/e, Pearson Education
5. ShibiParikkar, “ C# with .Net Frame Work” , Firewall Media.
6. Andrew Troelsen, —Pro C# 10 and the .NET 4 Platform, 5th Edition, Apress, 2010.
7. Bill Evjen, Scott Hanselman, Devin Rader, —Professional ASP.NET 3.5: In C# and VB (Programmer to Programmer), Wrox Publications, 2008.
8. Eric Butow and Tommy Ryan, —C# Your Visual Blueprint for building .NET Applications, Visual Publications, 2001.
9. Danny Ryan and Tommy Ryan, —ASP.NET Your Visual Blueprint for building Web on the .NET Framework, Visual Publisher, 2001.
10. Peter Wright, Beginning Visual C# 2005 Express Edition: From Novice to Professional, Apress, 2006.
11. Christian Nagelet et al, Professional C# 2005, Wiley Publishing, 2006.
12. Fritz Onion, Keith Brown, Essential ASP.NET 2.0, Addison Wesley , 2006.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER V	MCA 305	ARTIFICIAL INTELLIGENCE	L	T	P	C
			3	1	3	4
GOAL	To provide the basic exposition to goals and methods of Artificial Intelligence					
OBJECTIVES		OUTCOMES				
The course should enable the students to 1. Understand the representation of agents and agent environments. 2. Understand the searching techniques 3. Know the knowledge representation and learning 4. Enable the students to apply these techniques in application which involve perception, reasoning and learning 5. Know the features of expert systems		The student should be able to 1. Develop a basic understanding of the building blocks of AI 2. Understand the main approaches to artificial intelligence such as heuristic search, game and search. 3. Understand machine learning, neural networks and natural language processing. 4. Recognize problems that may be solved using artificial intelligence and implement artificial intelligence algorithms for hands-on experience. 5. Develop expert systems for an application.				

UNIT I:

Introduction: Introduction to Artificial Intelligence, Foundations and History of Artificial Intelligence, Applications of Artificial Intelligence, Intelligent Agents, Structure of Intelligent Agents. Computer vision, Natural Language Possessing, structure of agents, Problem Solving, problem solving agents, example problems, searching for solutions, uniformed search strategies, avoiding repeated states, searching with partial information.

UNIT II:

Introduction to Search : Searching for solutions, Uniformed search strategies, Informed search strategies, Local search algorithms and optimistic problems, Constraint satisfaction problems (CSP), Backtracking search and Local search for CSP , Structure of problems, Adversarial Search, Search for games, Alpha - Beta pruning, imperfect real-time decision, games that include an element of chance.

UNIT III:

Knowledge Representation & Reasoning: Propositional logic, Theory of first order logic, Inference in First order logic, Forward & Backward chaining, Resolution, Probabilistic reasoning, Utility theory, Hidden Markov Models (HMM), Bayesian Networks.

Applications: Communication, Communication as action, Formal grammar for a fragment of English, Syntactic analysis, Augmented grammars, Semantic interpretation, Ambiguity and disambiguation, Discourse understanding, Grammar induction, Probabilistic language, processing, Probabilistic language models, Information retrieval, Information Extraction, Machine translation.

UNIT IV:

Machine Learning : Supervised and unsupervised learning, Decision trees, Statistical learning models, Learning with complete data - Naive Bayes models, Learning from observations, forms of learning, Inductive learning, Learning decision trees, Ensemble learning, Knowledge in learning, Logical formulation of learning, Explanation based learning, Learning using relevant information, Inductive logic programming, Statistical learning methods, Learning with complete data, Learning with hidden data, EM algorithm, Reinforcement learning, Passive reinforcement learning, Active reinforcement learning, Generalization in reinforcement learning.

UNIT V:

Pattern Recognition : Introduction, Design principles of pattern recognition system, Statistical Pattern recognition, Parameter estimation methods - Principle Component Analysis (PCA) and Linear

Discriminant Analysis (LDA), Classification Techniques – Nearest Neighbor (NN) Rule, Bayes Classifier, Support Vector Machine (SVM), K – means clustering.

Expert System: Definition, Features of an expert system, Organization, Characteristics, Prospector, Knowledge Representation in expert systems, Expert system tools, MYCIN, EMYCIN.

TEXT BOOKS

1. Donald A. Waterman, „A Guide to Expert Systems“, Pearson Education.

REFERENCES:

1. Stuart Russell, Peter Norvig, “Artificial Intelligence – A Modern Approach”, Pearson Education
2. Elaine Rich and Kevin Knight, “Artificial Intelligence”, McGraw-Hill
3. E Charniak and D McDermott, “Introduction to Artificial Intelligence”, Pearson Education
4. Dan W. Patterson, “Artificial Intelligence and Expert Systems”, Prentice Hall of India,
5. Nils J. Nilsson, —Artificial Intelligence: A new Synthesis, Harcourt Asia Pvt. Ltd., 2000.
6. George F. Luger, —Artificial Intelligence-Structures And Strategies For Complex Problem Solving, Pearson Education / PHI, 2002.
7. Janakiraman, K. Sarukesi, „Foundations of Artificial Intelligence and Expert Systems“, Macmillan Series in Computer Science.
8. W. Patterson, „Introduction to Artificial Intelligence and Expert Systems“, Prentice Hall of India, 2003

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 306 (i)	Elective – II AUTOMATA AND COMPILER DESIGN	L	T	P	C
			3	1	3	4
GOAL	1. To provide the knowledge of automata theory to solve the problem. 2. To understand, design and implement a lexical analyzer, parser, code generation, code optimization and program verification phases of compiler.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Introduce Formal Languages, Automata Theory. 2. Learn Abstract models of Computation and Computability, Computational complexities and NP – Completeness. 3. Gain knowledge in computational theory. 6. Understand the basic principles of the compiler, Compiler construction tools and lexical analysis. 7. Learn the Concept of Context Free Grammars, Parsing and various Parsing Techniques. 8. Learn the process of intermediate code generation. 9. Learn the process of Code Generation and various Code optimization techniques. 10. Understand need of Program verification and Lambda calculus in verification.			The student should be able to 1. Apply the theoretical concepts and techniques in designing finite automata. 2. Convert regular expressions to FA and minimize Automata. 3. Write context free Grammar and design PDA for the Grammar. 4. Design turing machine and identify recursively enumerable languages. 5. Define undecidability and identify class P and NP problems. 6. Differentiate the various phases of a compiler. 7. Apply parsing techniques and able to write Context Free Grammars for various languages. 8. Design the structure of intermediate code for various types of statements and expressions. 9. Design code generator and apply code optimization techniques. 10. Apply Lambda Calculus to verify programs.			

UNIT I:

Automata: Languages – Grammars – Types of grammars – Context free grammar - regular expression - Recognizing of patterns - finite automation (deterministic & non deterministic) Conversion of NDFA to DFA - Conversion of regular expression of NDFA – Thompson’s construction-minimization of NDFA –Derivation - parse tree – ambiguity, CNF, GNF, Context-Free Grammar (CFG), Mealy and Moore machine, push down automata, Equivalence of Pushdown automata and CFG, Deterministic Pushdown Automata, Pumping Lemma for CFL, Turing machine, Programming Techniques for TM, Undecidable problems about Turing Machine.

UNIT II:

Introduction: Compilers, Analysis of the source program, Phases of a compiler, Cousins of the Compiler, Grouping of Phases, Compiler construction tools

Lexical Analysis:- Lexical analysis- handles - token specification - design of lexical analysis (LEX) - Automatic generation of lexical analyzer - input buffering - A language for specifying lexical analyzers - implementation of lexical analyzer

UNIT III:

Syntax Analysis – Parsing: Definition - role of parsers - top down parsing - bottom-up parsing - Left recursion - left factoring - Handle pruning , Shift reduce parsing - operator precedence parsing – FIRST- FOLLOW- LEADING- TRAILING- Predictive parsing - recursive descent parsing. LR parsing – LR (0) items - SLR parsing – Canonical LR - LALR parsing - generation of LALR - Ambiguous grammars - error recovery

UNIT IV:

Syntax Directed Translation: Intermediate Languages - prefix - postfix - Quadruple - triple - indirect triples – syntax tree- Evaluation of expression - three-address code- Synthesized attributes – Inherited attributes – Conversion of Assignment statements- Boolean expressions –Backpatching - Declaration - CASE statements.

UNIT V:

Code Optimization: Local optimization- Loop Optimization techniques – DAG – Dominators- Flow graphs – Storage allocations- Peephole optimization – Issues in Code Generation.

Program Verification: Introduction to Program Verification-Functional programming-Application of Lambda Calculus

Text Books

1. Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D. (2013). Introduction to Automata Theory, Languages, and Computation (3rd ed.). Pearson. ISBN 1292039051
2. Peter Linz, —An Introduction to Formal Language and Automata, Third Edition, Narosa Publishers, New Delhi, 2002

Reference

1. John C Martin, —Introduction to Languages and the Theory of Computation, Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007
2. Kamala Krithivasan and Rama. R, —Introduction to Formal Languages, Automata Theory and Computation, Pearson Education 2009
3. Alfred V Aho , Jeffery D Ullman , Ravi Sethi, " Compilers , Principles techniques and tools ", Pearson Education 2011
4. Raghavan V., “Principles of Compiler Design”, Tata McGraw Hill Education Pvt. Ltd., 2010.
5. David Galles, “Modern Compiler Design”, Pearson Education, Reprint 2012.
6. Dasaradh Ramaiah. K., “Introduction to Automata and Compiler Design”, PHI, 2011
7. Allen I. Holub —Compiler Design in C, Prentice Hall of India, 2003.
8. C.N. Fischer and R.J.LeBlanc, —Crafting a compiler with C, Benjamin Cummings, 2003.
9. J.P.Bennet, —Introduction to Compiler Techniques, 2nd Edition, Tata McGraw-Hill, 2003.
10. Henk Alblas and Albert Nymeyer, —Practice and Principles of Compiler Building with C, Prentice Hall, 2001.
11. Kenneth C. Loudon, —Compiler Construction: Principles and Practice, Thompson Learning, 2003.
12. Applications of Lambda calculus,
<http://wwwusers.di.uniroma1.it/~vamd/TSL/typedlambdacalculi.pdf>
13. Techniques for Program Verification www.eecs.berkeley.edu/~necula/Papers/nelsonthesis.pdf

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 306 (ii)	Elective – II DATABASE SECURITY	L	T	P	C
			3	1	3	4
GOAL	To learn Database security that concerns the use of a broad range of information security controls to protect databases, the database applications or stored functions, the database systems, the database servers and the associated network links against compromises of their confidentiality, integrity and availability.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Realize the issues in database security. 2. Learn database security models. 3. Study Security Mechanisms. 4. Learn the design of secure DBMS. 5. Learn statistical Database Protection system and IDS.			The student should be able to 1. Have understood the database security models. 2. Implement security mechanisms for database. 3. Design a secure DBMS. 4. Design IDS for DBMS. 5. Have learnt the models for new generation DBMS.			

UNIT I:

Introduction: Introduction to Databases Security, Problems in Databases Security, Controls Conclusions Security Models -1: Introduction, Access Matrix Model, Take-Grant Model, Acten Model, PN Model, Hartson and Hsiao's Model, Fernandez's Model, Bussolati and Martella's Model for Distributed databases.

UNIT II:

Security Models And Mechanisms: Bell and LaPadula's Model, Biba's Model, Dion's Model, Sea View Model, Jajodia and Sandhu's Model, The Lattice Model for the Flow Control conclusion. Security Mechanisms: Introduction, User Identification/Authentication, Memory Protection, Resource Protection, Control Flow Mechanisms, Isolation Security Functionalities in Some Operating Systems, Trusted Computer System, Evaluation Criteria.

UNIT III:

Security Software Design: Introduction, A Methodological Approach to Security Software Design, Secure Operating System Design, Secure DBMS Design, Security Packages, Database Security Design.

UNIT IV:

Statistical Database Protection & Intrusion Detection Systems: Introduction, Statistics Concepts and Definitions, Types of Attacks, Inference Controls evaluation, Criteria for Control Comparison, Introduction IDES System, RETISS System, ASES System Discovery.

UNIT V:

Models For The Protection Of New Generation Database Systems: Introduction, A Model for the Protection of Frame Based Systems, A Model for the Protection of Object-Oriented Systems, SORION Model for the Protection of Object-Oriented Databases, A Model for the Protection of New Generation Database Systems: the Orion Model Jajodia and Kogan's Model, A Model for the Protection of Active Databases Conclusions.

Text Books

1. Database Security and Auditing, Hassan A. Afyouni, India Edition, CENGAGE Learning, 2009.
2. Database Security, Castano, Second edition, Pearson Education.

Reference 1. Database Security, Alfred Basta, melissa zgola, CENGAGE learning.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 306 (iii)	Elective – II CRYPTOGRAPHY AND NETWORK SECURITY	L	T	P	C
			3	1	3	4
GOAL	To understand the principles of encryption algorithms, conventional and public key cryptography, detailed knowledge of authentication, hash functions and application level security mechanisms.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Know the methods of conventional encryption 2. Understand the concepts of number theory and its application in public key cryptography. 3. Understand authentication and Hash Functions. 4. Know the network security tools and application. 5. Understand the system level security used.			The student should be able to 1. Identify and classify computer and security threats and develop a security model to prevent, detect and recover from attacks. 2. Encrypt and decrypt messages using block ciphers. 3. Demonstrate techniques to Sign and verify messages using well-known signature generation and verification algorithms. 4. Develop code to implement a cryptographic algorithm or write an analysis report on any existing security product. 5. Understand and demonstrate the technologies to protect cipher space against security threats.			

UNIT I:

Introduction: OSI Security Architecture, Classical Encryption techniques, Cipher Principles, Data Encryption Standard, Block Cipher Design Principles and Modes of Operation, Evaluation criteria for AES, AES Cipher, Triple DES, Placement of Encryption Function, Traffic Confidentiality

UNIT II:

Public Key Cryptography: Key Management, Diffie-Hellman key Exchange, Elliptic Curve Architecture and Cryptography, Number Theory concepts, Confidentiality using Symmetric Encryption, Public Key Cryptography and RSA.

UNIT III:

Authentication And Hash Function: Authentication requirements, Authentication functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and MACs, MD5 message Digest algorithm, Secure Hash Algorithm, RIPEMD, HMAC Digital Signatures, Authentication Protocols, Digital Signature Standard.

UNIT IV:

Network Security: Authentication Applications: Kerberos, X.509 Authentication Service, Electronic Mail Security, PGP, S/MIME, IP Security, Web Security.

UNIT V:

System Level Security:

Intrusion detection, password management, Viruses and related Threats, Virus Counter measures, Firewall Design Principles, Trusted Systems.

Text Book

1. William Stallings, Cryptography And Network Security, Principles and Practices, Prentice Hall of India, Fourth Edition, 2010

Reference

1. Atul Kahate, Cryptography and Network Security, Tata McGraw-Hill, 2003.
2. Bruce Schneier, Applied Cryptography, John Wiley & Sons Inc, 2001.
3. Charles B. Pfleeger, Shari Lawrence Pfleeger, Security in Computing, Third Edition, Pearson Education, 2003.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 306 (iv)	Elective – II SOFTWARE AND APPLICATION SECURITY	L	T	P	C
			3	1	3	4
GOAL	To impart knowledge on web application security and to develop a secure application through secure coding					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Understand the security fundamentals 2. Learn Network Security Programming 3. Learn various security attacks. 4. Understand web application security 5. Learn secure coding in C, C++ and Java			The student should be able to 1. possess the knowledge of the fundamental concepts of security 2. Securely program servers and clients 3. Capture packets and analyze the packets for security attacks 4. Apply web application attacks 5. Develop a secure code.			

UNIT I:

Security Fundamentals: Security Attacks, Security Services, Security Mechanisms, Need for secure systems- Proactive security development process- Security principles, threat modelling.

UNIT II:

Network Security Programming: Raw Socket basics, Socket Libraries and Functionality, Programming Servers and Clients, Programming Wired and Wireless Sniffers, Programming arbitrary packet injectors, PCAP file parsing and analysis.

UNIT III:

Web Application Security: Web Servers and Client scripting, Web Application Fuzzers, Scraping Web Applications, HTML and XML file analysis, Web Browser Emulation, Attacking Web Services, Application Proxies and Data Mangling, Automation of attacks such as SQL Injection, XSS.

UNIT IV:

Security Coding in C: Character strings- String manipulation errors, String Vulnerabilities and exploits, Mitigation strategies for strings- Pointers, Mitigation strategies in pointer based vulnerabilities, Buffer Overflow based vulnerabilities.

UNIT V:

Security Coding In C++ and Java: Dynamic memory management- Common errors in dynamic memory management- Memory managers- Double –free vulnerabilities –Integer security- Mitigation strategies

Reference

1. William Stallings, " Cryptography and Network Security" , 5th Ed., Pearson, 2010.
2. Michael Howard , David LeBlanc, —Writing Secure Code, Microsoft Press, 2nd Ed., 2003
3. Robert C.Seacord, —Secure Coding in C and C++, Pearson Education, 2nd Ed., 2013

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 306 (v)	Elective – II NEURAL NETWORK AND FUZZY LOGIC	L	T	P	C
			3	1	3	4
GOAL	To provide knowledge about Neural Network and Fuzzy Logic.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Learn the concepts of fuzzy sets, fuzzy logic and heuristics based on human experience. 2. Understand neural network techniques. 3. Learn the mathematical background for carrying out optimization associated with neural network learning. 4. Learn genetic algorithms and random search procedures. 5. Learn the methodologies to illustrate the intelligent behaviour of programs based on soft computing 6. To perform such cognitive functions as problem solving and machine learning.			The student should be able to 1. Apply concepts of fuzzy sets, fuzzy logic and heuristics based systems. 2. Derive appropriate rules for inference systems. 3. Have understood the mathematical background to optimize neural network learning. 4. Implement optimization algorithms and random search procedures useful to seek global optimum in self-learning. 5. Develop case studies to illustrate the intelligent behaviour of programs based on soft computing 6. Apply neural networks for suitable application. 7. To develop investment decisions, recognize handwriting, and even detect bombs. 8. To recognize correlations and hidden patterns in raw data and also to cluster and classify raw data and to continuously learn and improve over time.			

UNIT I:

Neural Networks Characteristics: History of Development in neural networks, Artificial neural net terminology, model of a neuron, Topology, Types of learning. Supervised, Unsupervised learning. Basic Learning laws, Hebb's rule, Delta rule, widrow and Hoff LMS learning rule, correlation learning rule instar and ouster learning rules.

UNIT II:

Unsupervised Learning: Competitive learning, K-means clustering algorithm, Kohonen's feature maps. Radial Basis function neural networks- recurrent networks, Real time recurrent and learning algorithm. Introduction to Counter propagation Networks- CMAC Network, ART networks, Application of NN in pattern recognition, optimization, Control, Speech and decision making.

UNIT III:

Single Layer Feed Forward Neural Networks Introduction, Perceptron Models: Discrete, Continuous and Multi-Category, Training Algorithms: Discrete and Continuous Perceptron Networks, Perceptron Convergence theorem, Limitations of the Perceptron Model, Applications.

UNIT IV:

Multilayer Feed forward Neural Networks Credit Assignment Problem, Generalized Delta Rule, Derivation of Backpropagation (BP) Training, Summary of Backpropagation Algorithm, Kolmogorov Theorem, Learning Difficulties and Improvements.

UNIT V:

Fuzzy Logic: Basic concepts of Fuzzy logic, Fuzzy vs Crisp set, Linguistic variables, membership functions, operations of Fuzzy sets, Fuzzy if-then rules, Fuzzy Rules and Fuzzy Reasoning, Variables inference techniques, defuzzification techniques, basic Fuzzy inference algorithm, application of fuzzy logic , Fuzzy system design implementation , useful tools supporting design.

Neuro Fuzzy Modeling: Adaptive Neuro, Fuzzy Inference Systems , Architecture , Hybrid Learning Algorithm , Learning Methods that Cross-fertilize ANFIS and RBFN , Coactive Neuro Fuzzy Modeling , Framework Neuron Functions for Adaptive Networks , Neuro Fuzzy Spectrum.

Text Books :

1. Berkin Riza C and Trubatch, “ Fuzzy System design principles- Building Fuzzy IF-THEN rule bases”, IEEE Press.
2. Yegna Narayanan, “Artificial Neural Networks”. 8th Printing. PHI(2003)
3. Patterson Dan W, “Introduction to artificial Intelligence and Expert systems”, 3rd Ed., PHI
4. Simon Haykin, “Neural Networks” Pearson Education.
5. Yen and Langari, “Fuzzy Logic: Intelligence, Control and Information”, Pearson Education.
6. Jacek M Zaurada, “Introduction to artificial neural Networks Jaico Publishing Home, Fouth Impression.

PAPER NO.	COURSE CODE	COURSE TITLE	PERIODS			
PAPER VI	MCA 306 (vi)	Elective – II WIRELESS SENSOR NETWORKS	L	T	P	C
			3	1	3	4
GOAL	Introduce the students to the diverse literature on sensor networks, and expose them to the fundamental issues in designing and analyzing sensor network					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. Describe the current technology trends and unique issues for the implementation in sensor networks 2. Learn the concept of MAC protocols the sensor networks 3. Learn routing strategies and design issues in WSN 4. Understand the various tools and programming challenges for simulating a environment for sensor systems using Motes. 5. Understand design issues performance and traffic management of WSN			The student should be able to 1. Explain the basic wireless sensor technology and its applications. 2. Explain the MAC protocols of WSN. 3. Explain the challenges in designing routing protocols and routing techniques in WSN 4. Understand the operating system of WSN and its components 5. Explain the working models and performance of a WSN.			

UNIT I:

Introduction and Overview of Wireless Sensor Networks: Introduction, Brief Historical Survey of Sensor Networks, and Background of Sensor Network Technology, Ad-Hoc Networks, Applications of Wireless Sensor Networks: Sensor and Robots, Reconfigurable Sensor Networks, Highway Monitoring, Military Applications, Civil and Environmental Engineering Applications, Wildfire Instrumentation, Habitat Monitoring, Home Control, Basic Wireless Sensor Technology : Introduction, Sensor Node Technology, Sensor Taxonomy, WSN Operating Environment, WSN Trends, Wireless Network Standards: IEEE 802.15.4, ZigBee.

UNIT II:

Medium Access Control Protocols for Wireless Sensor Networks: Fundamentals of MAC Protocols, MAC Protocols for WSNs: Schedule-Based Protocols, Random Access-Based Protocols.

UNIT III:

Routing Protocols for Wireless Sensor Networks: Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless Sensor Networks Network Scale and Time-Varying Characteristics, Resource Constraints, Sensor Applications Data Models, Routing Strategies in Wireless Sensor Networks: WSN Routing Techniques, Flooding and Its Variants, Sensor Protocols for Information via Negotiation, Low-Energy Adaptive Clustering Hierarchy, Power-Efficient Gathering in Sensor Information Systems, Directed Diffusion, Geographical Routing,

UNIT IV:

Operating Systems for Wireless Sensor Networks: Operating System Design Issues, Example of Operating Systems,(TinyOS), Introduction to Tiny OS – NesC – Interfaces and Modules-Configurations and Wiring - Generic Components .

UNIT V:

Performance and Traffic Management of Wireless Sensor Networks: WSN Design Issues, MAC Protocols, Routing Protocols, Transport Protocols, Performance Modeling of WSNs, Performance Metrics, Basic Models, Network Models,

Text Books

1. Kazem Sohraby, Daniel Minoli and Taieb Znati, Wireless Sensor Networks Technology, Protocols, and Applications, John Wiley & Sons, 2007.

Reference

- 1.K. Akkaya and M. Younis, A survey of routing protocols in wireless sensor networks, Elsevier Ad Hoc Network Journal, Vol. 3, no. 3, pp. 325--349
- 2.Philip Levis, — TinyOS Programming
- 3.Anna Ha'c, —Wireless Sensor Network Designs, John Wiley & Sons Ltd,
4. Holger Karl and Andreas Willig, —Protocols and Architectures for Wireless Sensor Networks, John Wiley & Sons, Ltd, 2005.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB I	MCA 307	ANDROID PROGRAMMING LAB	L	T	P	C
			0	0	3	2
GOAL	To impart knowledge on various aspects of Android programming					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To have a review on concept of Android programming. 2. To learn Android Programming Environments. 3. To practice programming in Android. 4. To learn GUI Application development in Android platform with XML			The student should be able to 1. Understand the various Android programming architectures. 2. Design and develop mobile apps, using Android as development platform, with key focus on user experience design, native data handling and background tasks and notifications. 3. Write mobile applications using J2ME.			

List of Android Programming Experiments:

- Lab 1: Programs to understand basic arithmetic operations
- Lab 2: Programs to understand basic logic operations
- Lab 3: Programs to understand loops and control statements
- Lab 4: Programs to understand GUI in android
- Lab 5: Android application for adding two numbers
- Lab 6: Develop simple user interface to display message
- Lab 7: Create two menu items-opening a file-saving a file
- Lab 8: Text view controls to represent each row in a list view
- Lab 9: Implementation of background image
- Lab 10: Starting another activity from your own activity using intent
- Lab 11: Create a new activity that services ACTION-PICK for contact data which display each of the contact in the contact database and lets the user to select one before closing and returning the selected contacts URL to the calling activities
- Lab 12: Create Android application to linkify a text view to display web and E-mail address as hyperlinks. When clicked they will open the browser and E-mail address respectively
- Lab 13: Implementation of array adapter
- Lab 14: Create an alert dialogs used to display a message and offer two button options to continue. Clicking either button will close the dialog after executing the attached click listener
- Lab 15: Create an earth quake viewer
- Lab 16: Create mobile applications
- Lab 17: Program to implement simple calculator
- Lab 18: Program to Get IP Address
- Lab 19: Program to Home And Lock Screen Widget (Temperature Widget)
- Lab 20: Program to Device/Battery Temperature Sensor
- Lab 21: Program to Audio Demo AudioTrack , AudioRecord
- Lab 22: Program to Blocking Incoming call Android
- Lab 23: Program to create simple login screen.
- Lab 24: Implementing communication commands
- Lab 25: Implementation of managing user accounts
- Lab 26 Implementing backup and restore
- Lab 27: Configuration of web servers and proxy server
- Lab 28: Configuration of DNS servers and mail server

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB II	MCA 308	PHP PROGRAMMING LAB	L	T	P	C
			0	0	3	2
GOAL	The Objective is to familiarize students with the latest trends and advancements in PHP programming Languages which are used in industries					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To learn client side and server side scripting. 2. To learn PHP scripts to handle HTML forms. 3. To learn PHP Programming. 4. To learn how to develop static and dynamic websites. 5. To learn how to interact with databases through internet.			The student should be able to 1. To develop client side and server side scripting. 2. Write PHP scripts to handle HTML forms. 3. To develop PHP Programming. 4. To develop developing Static and dynamic websites. 5. To develop how to interact with databases through PHP.			

List of PHP Programming experiments

Lab 1: Setting-up the environment

Setup WAMP/XAMPP Server or Setup Apache, MySQL and PHP separately in your PHP Lab.

Simple PHP program that displays a welcome message.

Write a php program to generate a random number between 1 and 100.

Modify above program to accept range of the random number from HTML interface.

Lab 2: Programs involving various control structures like:

‘if, else, elseif/else if’

Alternative Syntax for ‘if, else, elseif/else if’

Lab 3: Programs involving various control structures like:

while, do-while, for, foreach, switch, break, continue.

Try alternative syntax for while, do-while, for, foreach, switch.

Lab 4: Programs involving the following.

declare, return.

require, include, require- once, include_once and goto.

Lab 5: Programs to demonstrate PHP Array functions.

PHP Array Sorting,

PHP Key Sorting,

PHP Value Sorting,

PHP MultiArray Sorting,

PHP Array Random Sorting,

Lab 6: Programs to demonstrate PHP Array functions.

PHP Array Reverse Sorting,

Array to String Conversion,

Implode() function,

String to Array, Array Count,

Remove Duplicate Values

Lab 7: Programs to demonstrate PHP Array functions.

Array Search,
Array Replace,
Array Replace Recursive,
Array Sub String Search

Lab 8: Demonstrate the following.

Use of regular expression to compare two strings.
Extract Domain name from URL.
Find the number of rows from a mysql database for your query.

Lab 9: Generate a Guestbook which will allow your website visitor to enter some simple data about your website.

Lab 10: Develop a PHP program for Email Registration.

Lab11: Develop a project for making Application form and performing Degree Admission On-line.

Write PHP program on the following:

1. PHP Basic
2. PHP arrays
3. PHP for loop
4. PHP functions
5. PHP classes
6. PHP Regular Expression
7. PHP Date
8. PHP String
9. PHP Math
10. PHP JSON
11. PHP Searching and Sorting Algorithm

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB III	MCA 309	COMPUTER NETWORK LAB	L	T	P	C
			0	0	3	2
GOAL	To simulate the various protocols, develop various applications and study the various network simulators.					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To provide students with a theoretical and practical base in computer networks issues 2. Student will be able pursue his study in advanced networking courses 3. Enable the students to establish basic network connection using TCP/IP protocol. 4. Learning networking concepts through D-Link			The student should be able to 1. Implement client and server concepts in Network system. 2. Implement Error correction method in data in flow 3. Implement the different protocols. 4. Implement the concept of local area networks, their topologies, protocols			

LIST OF EXPERIMENTS

(All the programs are to be written using C/ C++/ NS2 /NS3 / python)

<https://www.nsnam.org/>

- Simulation of ARP / RARP.
- Write a program that takes a binary file as input and performs bit stuffing and CRC Computation.
- Develop an application for transferring files over RS232.
- Simulation of Sliding-Window protocol.
- Simulation of BGP / OSPF routing protocol.
- Develop a Client – Server application for chat.
- Develop a Client that contacts a given DNS Server to resolve a given host name.
- Write a Client to download a file from a HTTP Server.
- Study of Network Simulators like NS2/Glomosim / OPNET
- To access and configure the Switch for basic Switch operations.
- Creating static V LAN and configuring Ports. 12. To configure routing using two different methods: static and dynamic.
- To understand the fundamentals of networking and the TCP/IP protocol suite to be learnt.
- To understand the fundamentals of networking and the TCP/IP protocol suite to be learnt.
- To access and configure the Switch for basic Switch operations.
- To create and configure VLANs on the switch and static and Dynamic (GVRP) VLANs
- To create and configure a Spanning Tree Protocol (STP).
- To configure stacking using two different methods: physical and virtual. 19. To configure routing using two different methods: static and dynamic.
- To configure DHCP, ACL, LLDP, and System Maintenance.
- To learn the topologies for the basic WLAN Design
- To learn the topology in the basic metropolitan area design
- To configure two SSIDs and apply small business and teleworker security
- To configure WPA2 PSK and WPA2-EAP Authentication on unified Access points
- To configure an Air premier NAP for WDS with AP mode
- To perform the IP camera installation ,configuration and the settings of Image setup, motion detection, recording
- To configure D-View Cam software application
- To configure the network Video Recorder (NVR) for basic operations

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB IV	MCA 310	DOT NET FRAMEWORK & C# LAB	L	T	P	C
			0	0	3	2
GOAL	To learn and develop the fundamental concepts in .NET framework and programming.					
OBJECTIVES		OUTCOMES				
The course should enable the students to 1. Gain knowledge in the concepts of the .NET framework as a whole and the technologies that constitute the framework. 2. Write Programs in C# and ASP.Net, both in basic and advanced levels. 3. Build sample applications. 4. Develop an appreciation of key generic .Net concepts and techniques. 5. To learn the basics of C# programming using .NET		The student should be able to 1. To develop the basis of concept .Net framework. & C# programming 2. Understand object oriented Aspects of C# and ASP.Net. 3. Develop simple applications under .Net framework. 4. Develop Web based Applications using .Net programming languages.				

Write programs in C# illustrating

1. The use of sequence, conditional and iteration construct, different control structures
2. Various operators like logical, arithmetical, relational, etc.
3. Overloading of various operators.
4. Use of Fried, Inline, and Static Member functions, default arguments.
5. Use of destructor and various types of constructor.
6. Various forms of Inheritance.
7. Use of virtual functions, virtual Base Class, delegates.
8. File operation.
9. Simple web application using ASP Net.
10. Use of Active X controls.
11. Simple c# .NET console applications
12. Simple C# .NET Windows applications
14. Programs involving various array operations
15. Programs involving various string functions
16. Programs involving forms and multiple forms
17. Programs involving mouse and keyboard events
18. Programs involving MsgBox, InputBox
19. Programs involving classes and objects
20. Programs involving exception handling
21. Programs involving common controls
22. Programs involving file controls
23. Programs involving OLE
24. Programs involving data access using ADO.NET

Note : Students are advised to develop a small project illustrating the handling of database and screens in order to fully understand the C#.

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB V	MCA 311	ARTIFICIAL INTELLIGENCE LAB	L	T	P	C
			0	0	3	2
GOAL	To provide the basic exposition to goals and methods of Artificial Intelligence Programming					
OBJECTIVES			OUTCOMES			
The course should enable the students to 1. To learn Artificial Intelligence, its foundation and principles. 2. Learn the practical applicability of intelligent systems, specifically its applications. 3. Learn programming language to program intelligent systems. 4. Understand the searching techniques 5. Know the knowledge representation and learning 6. Enable the students to apply programs in application which involve perception, reasoning and learning			The student should be able to 1. Develop a basic understanding of the building blocks of AI 2. Understand the main approaches to artificial intelligence such as heuristic search, game and search. 3. Be familiar with Artificial Intelligence, its foundation and principles. 4. Examine the useful search techniques; learn their advantages, disadvantages and comparison. 5. To develop programming language to program intelligent systems. 6. Understand important concepts like Expert Systems, AI applications. 7. Be exposed to the role of AI in different areas like NLP, Pattern Recognition etc. 8. tO develop the practical applicability of intelligent systems, specifically its applications. 8. Be able to develop intelligent systems.			

1. Study of facts, objects, predicates and variables in PROLOG.
2. Write simple fact for the statements using PROLOG.
3. Study of Rules and Unification in PROLOG.
4. Study of “cut” and “fail” predicate in PROLOG.
3. Write predicates One converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.
4. Write a program to solve the Monkey Banana problem.
5. Study of recursion in PROLOG.
6. Study of Lists in PROLOG.
7. Study of dynamic database in PROLOG.
8. Study of string operations in PROLOG. Implement string operations like substring, string position, palindrome etc.)
9. Write a prolog program to maintain family tree.
10. Write a prolog program to implement all set operations (Union, intersection, complement etc.)
11. Write a prolog program to implement Library Management system.
12. Write a prolog program to solve “Water Jug Problem”
13. Study of arithmetic operators, simple input/output and compound goals in PRO LOG.
14. WAP in turbo prolog for medical diagnosis and show the advantage and disadvantage of green and red cuts.
15. WAP to implement factorial, fibonacci of a given number.
16. Write a program to solve 4-Queen problem.
17. Write a program to solve traveling salesman problem.
18. Write a program to solve water jug problem using LISP

LAB NO.	COURSE CODE	COURSE TITLE	PERIODS			
LAB VI	MCA 112	PROJECT / SEMINAR*	L	T	P	C
			0	0	3	2
GOAL	1. The project goal refers to achieving a desired outcome (Theory and Practical knowledge of on new technology) at a specific end date employing a specific amount of resources. 2. Seminar may be for the purpose of education, such as a lecture, where the participants engage in the discussion of an academic subject for the aim of gaining a better insight into the subject.					
OBJECTIVES			OUTCOMES			
The course should enable the students to: 1. A project objective is a statement that describes the “what” of your project. The tangible and measurable “what”. The “what” that's achievable, realistic, and can be completed within the time allowed. These statements ladder up to the goals of the project, providing stepping stones to success. What should knowledge gain from Industry internship / training (45 days) after end MCA IIND SEM 2. To provide a comprehensive understanding of the Basic knowledge of the Industry internship on new technology 3. To encourage the use of the knowledge by PPT 4. To promote positive relations between Faculty members and Classmates 5. To develop informed and responsibility			At the Industry internship end of the course the student should be able to: 1. The value of any project cannot be measured without defining success. Outcomes are specific, measurable, and meaningful. knowledge gain from Industry internship / training (45 days) after end MCA IIND SEM 2. Student Learning Outcomes from Seminar i. Presentation Skills. ii. Discussion Skills. iii. Listening Skills. iv. Argumentative Skills and Critical Thinking. Questioning. v. Interdisciplinary Inquiry. vi. Engaging with Big Questions. vii. Studying Major Works.			

End of MCA 2nd Semester Exam students will do 45 days Industry Internship/ training which is based on Latest technology.

Students make a project thesis based on Industry Internship / training which you done after IIND Sem End Semester Examination. Maximum Two Students may be participated in one Project. After developing Project thesis student presenting an explanation on a specified concept may include seminar in it. Students develop project under the guidance of faculty members / Industry Expert. Student Submitted Industry Internship/ training Certificate with Project Thesis.

Prepare Project Thesis according Appendix 1

Dr. RamManohar Lohia Avadh University, Ayodhya U.P.
Study and Evaluation Scheme
MCA (Master of Computer Applications)
(Effective From Session 2021-22)
Year – II Semester – IV

Sl. No.	Subject Code	Subject Name	Periods			Evaluation Scheme					Credit
			L	T	P	Sessional Exams			ESE	Subject Total	
						CT	TA	Total			
	THEORY SUBJECT										
1	MCA 401	Big Data	3	1	0	30	20	50	100	150	04
2	MCA 402	Python Programming	3	1	0	30	20	50	100	150	04
3	MCA 403	Data Science	3	1	0	30	20	50	100	150	04
4	MCA 404	Machine Learning	3	1	0	30	20	50	100	150	04
5	MCA 405	Data Mining and warehousing	3	1	0	30	20	50	100	150	04
6	MCA 406	Elective – III	3	1	0	30	20	50	100	150	04
	Practical										
7	MCA 407	Big Data Lab	0	0	3	10	10	20	30	50	02
8	MCA 408	Python Programming Lab	0	0	3	10	10	20	30	50	02
9	MCA 409	Data Science Lab	0	0	3	10	10	20	30	50	02
10	MCA 410	Machine Learning Lab	0	0	3	10	10	20	30	50	02
11	MCA 411	Data Mining and warehousing Lab	0	0	3	10	10	20	30	50	02
12	MCA 412	Dissertation*	0	0	3	60	40	100	150	250	02
		Total	18	6	18	-	-	-	-	1400	36

Elective – III

MCA 406-(i)	Distributed Architecture of Enterprise Applications
MCA 406-(ii)	Parallel Processing
MCA 406-(iii)	Forensic Science and Application
MCA 406-(iv)	Digital Image Processing
MCA 406-(v)	Cloud computing
MCA 406-(vi)	Grid Computing

Students make a project thesis in specific topic on Research area / Specialization based on latest technology. Maximum Two Students may be participated in one Project. After developing Project thesis student presenting an explanation on a specified concept may include seminar in it. Students develop project under the guidance of faculty members. Students must publish at least one paper in UGC care list / SCI index / Scopus index / Web science.

Prepare Project Thesis according MCA Thesis Guidelines 2020

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER I	MCA 401	BIG DATA	100	L	T	P	C
				3	1	3	4
GOAL	To increase the speed at which products get to market, to reduce the amount of time and resources required to gain market adoption, target audiences, and to ensure customers remain satisfied.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Gain knowledge in Big Data. 2. Understand the concepts and applications of Big Data. 3. An organisation or individual can obtain, store, transform and analyse large amounts of data to solve specific problems. 4. A data driven approach to understanding a business. One can build predictive data models and detect future trends. Devices of the future will be built entirely on big data.			The student should be able to 1. Understanding of Big Data 2. Implement various NoSQL Data Management 3. Develop applications using Hadoop 4. Implement Map Reduce for different applications. 5. Implement Hadoop related tools 6. Using big data helps you increase sales and loyalty, increases your efficiency, improves your pricing & ensures you hire the right employees.				

UNIT I UNDERSTANDING:

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

UNIT II NOSQL DATA MANAGEMENT

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

UNIT III BASICS OF HADOOP

Data format, analyzing data with Hadoop, scaling out, Hadoop streaming, Hadoop pipes, design of Hadoop distributed file system (HDFS), HDFS concepts, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization, Avro file-based data structures

UNIT IV MAP REDUCE APPLICATIONS

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

UNIT V HADOOP RELATED TOOLS

Hbase, data model and implementations, Hbase clients, Hbase examples – praxis. Cassandra, cassandra data model, cassandra examples, cassandra clients, Hadoop integration. Pig, Grunt, pig data model, Pig Latin, developing and testing Pig Latin scripts. Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation – HiveQL queries

Text Books:

1. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
2. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.
3. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012.
4. Eric Sammer, "Hadoop Operations", O'Reilley, 2012.
5. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.
6. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011.
7. Eben Hewitt, "Cassandra: The Definitive Guide", O'Reilley, 2010.
8. Alan Gates, "Programming Pig", O'Reilley, 2011.

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER II	MCA 402	PYTHON PROGRAMMING	100	L	T	P	C
				3	1	3	4
GOAL	To emphasizes programmer productivity and code readability. To build websites and software, automate tasks, and conduct data analysis.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Gain knowledge in Python programming. 2. Understand the concepts and applications of Python programming. 3. Build basic programs using fundamental programming constructs like variables, conditional logic, looping, and functions. 4. To provide problem solving and programming capability.			The student should be able to 1. Understanding of Python programming. 2. Build basic programs using conditional logic, looping, List and functions 3. Implement Database Access, GUI Programming for different applications. 4. Implement Networking, XML Processing, for different applications. 5. Develop applications using Python programming.				

UNIT I

Introduction: History, Features, Setting up path, working with Python, Basic Syntax, Variable and Data Types, Operator,

Conditional Statements: If, If- else, Nested if-else

Looping: For, While, Nested loops

UNIT II

Control Statements: Break, Continue, Pass, String Manipulation, Accessing Strings, Basic Operations, String slices, Function and Methods

Lists: Introduction, Accessing list, Operations, Working with lists, Function and Methods

Tuple: Introduction, Accessing tuples, Operations, Working, Functions and Methods

UNIT III

Dictionaries: Introduction, Accessing values in dictionaries, working with dictionaries, Properties, Functions

Functions: Defining a function, calling a function, Types of functions, Function Arguments, Anonymous functions, Global and local variables

UNITs: Importing UNIT, Math UNIT, Random UNIT, Packages, Composition,

UNIT IV

Input-Output: Printing on screen, Reading data from keyboard, Opening and closing file , Reading and writing files, Functions

Exception Handling: Exception, Exception Handling, Except clause, Try ? Finally clause, User Defined Exceptions

UNIT V

Advanced Python: Classes Objects, Reg Expressions, CGI Programming, Database Access, Networking, Sending Email, Multithreading, XML Processing, GUI Programming, Further Extensions

References:

1. Programming With Python Book,Himalaya Publishing House Pvt. Ltd.
2. Python for Data Analysis by Wes McKinney
3. Python for Data Analysis by O'Reilly
4. Python Programming: An Introduction to Computer Science by Jhon Zelly
5. Learn Python in one day and learn it well Jamie Chan

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER III	MCA 403	DATA SCIENCE	100	L	T	P	C
				3	1	3	4
GOAL	To extract, pre-process and analyze data and help solve problems. To make businesses grow better.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Gain knowledge in Data science. 2. Understand the concepts and applications of Data science. 3. To explore, sort and analyze mega data from various sources in order to take advantage of them and reach conclusions to optimize business processes or for decision support. 4. To construct the means for extracting business-focused insights from data.			The student should be able to 1. Understanding of data science. 2. Build Python programs for data science using functions, OOPs and file handling and numpy 3. Using Pandas Implement Data manipulation and Database for different applications. 4. Implement data visualization with MATPLOTLIB for different applications. 5. Develop Data Science applications using Python programming and Machine Learning				

Unit I INTRODUCTION TO DATA SCIENCE: Introduction to Data Science and its importance, Data Science and Big data, The life cycle of Data Science, The Art of Data Science, Work with data, Types of Data, Data privacy, Data Security and ethics, Tools for Data Science, Data Cleaning, Data Munging, Data manipulation. Establishing computational environments for data scientists using Python with IPython and Jupyter Notebook.

Unit II PYTHON FOR DATA SCIENCE-Basic Python concepts, Python Data structures, String Manipulation, Functions, Class, Object, Overloading, Overriding, Inheritance, Information hiding, Modules, Packages and File handling

NUMPY: The Basics of NumPy Arrays - Computation on NumPy Arrays: Universal Functions - Aggregations: Min, Max, and Everything in Between Computation on Arrays: Broadcasting-Comparisons, Masks, and Boolean Logic Fancy Indexing-Sorting Arrays

Unit III DATA MANIPULATION USING PANDAS: Installing and Using Pandas, Introducing Pandas Objects, Data Indexing and Selection. Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing Combining Datasets: Concat and Append, Combining Datasets: Merge and Join. Aggregation and Grouping, Pivot Tables, Vectorized String Operations, Working with Time Series.

INTRODUCTION TO DATABASES AND BASIC SQL: Introduction to Databases, how to create a Database instance, Relational Database Concepts, CREATE Table Statement, SELECT Statement, COUNT, DISTINCT, LIMIT, INSERT Statement, UPDATE and DELETE Statements.

ADVANCED SQL- Using String Patterns, Ranges, Sorting Result Sets, Grouping Result Sets, Built-in database Functions, Date and Time Built-in Functions, Sub-Queries and Nested Selects, Working with Multiple Tables, Types of Joins with Examples

Unit IV DATA VISUALIZATION WITH MATPLOTLIB

General Matplotlib Tips, Simple Line Plots, Simple Scatter Plots, Visualizing Errors Density and Contour Plots, Histograms, Binning and Density, Customizing Plot Legends Customizing Color bars, Multiple Subplots, Text and Annotation, Customizing Ticks Customizing Matplotlib: Configurations and Stylesheets, Geographic Data with Basemap.

Unit V MACHINE LEARNING USING PYTHON: Intro Machine Learning, Categories of Machine Learning algorithms, Dimensionality Reduction, Introducing Scikit-Application: Exploring Hand-written Digits. Feature Engineering- Naive Bayes Classification, Linear Regression, K-Nearest Neighbors (k-NN), SVM, k-Means Clustering, Decision Tree, Building Pipelines, Training Model, Saving Model, Loading Model, Writing API for Model using Flask, Deployment of Model

Text Book:

1. Python Data Science Handbook-Essential Tools for Working with Data, Jake Vander Plas, O'Reilly Media, 2016.
2. Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly, 2015.
3. SQL Cookbook, 2nd Edition, O'Reilly, 2020
4. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly,2020

Reference Book:

1. Python for Data Analysis, Wes McKinney, O'Reilly Media, 2013.
2. Field Cady, "Data Science Hand Book", John Wiley & Sons, 2017.
3. Fundamentals of Data Science, Samuel Burns, Amazon KDP printing and Publishing, 2019.
4. Doing Data Science, Straight Talk from The Frontline, Cathy O'Neil and Rachel Schutt. O'Reilly. 2014.
5. Tony Ojeda, Sean Patrick Murphy, Benjamin Bengfort, Abhijit Dasgupta, "Practical Data Science Cookbook", Packt Publishing Ltd., 2014.
6. Nathan Yau, "Visualize This: The Flowing Data Guide to Design, Visualization, and Statistics", Wiley, 2011.
7. Shai Vaingast, "Beginning Python Visualization Crafting Visual Transformation Scripts", Apress, 2nd edition, 2014.

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER IV	MCA 404	MACHINE LEARNING	100	L	T	P	C
				3	1	3	4
GOAL	To impart knowledge on Machine Learning						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. To understand the concepts of machine learning 2. To learn supervised and unsupervised learning and their applications 3. To understand the theoretical and practical aspects of Probabilistic Graphical Models 4. To appreciate the concepts and algorithms of reinforcement learning 5. To learn aspects of computational learning theory			The student should be able to 1. Upon Completion of the course, the students will be able to 2. Apply neural networks for suitable application. 3. Implement probabilistic discriminative and generative algorithms for an application of your choice and analyze the results 4. Use a tool to implement typical clustering algorithms for different types of applications 5. Design and implement an HMM for a sequence model type of application				

UNIT I INTRODUCTION

Machine Learning, Machine Learning Fundamentals – applications, Types of machine learning, basic concepts in machine learning , Examples of Machine Learning

UNIT II SUPERVISED LEARNING

Linear Models for Regression , Linear Basis Function Models , The Bias, Variance Decomposition , Bayesian Linear Regression , Bayesian Model Comparison , Linear Models for Classification , Discriminant Functions , Probabilistic Generative Models , Probabilistic Discriminative Models , Bayesian Logistic Regression. Decision Trees, Classification Trees, Regression Trees, Pruning. Neural Networks, Feed, forward Network Functions, Back, propagation

UNIT III UNSUPERVISED LEARNING

Clustering, K, means , EM , Mixtures of Gaussians , The EM Algorithm in General , Model selection for latent variable models , high dimensional spaces , The Curse of Dimensionality , Dimensionality Reduction , Factor analysis , Principal Component Analysis , Probabilistic PCA, Independent components analysis

UNIT IV PROBABILISTIC GRAPHICAL MODELS- I

Directed Graphical Models , Bayesian Networks , Exploiting Independence Properties , From Distributions to Graphs , Examples , Markov Random Fields , Inference in Graphical Models , Learning–Naive Bayes classifiers, Markov Models – Hidden Markov Models – Inference – Learning, Generalization

UNIT V PROBABILISTIC GRAPHICAL MODELS- II

Undirected graphical models, Markov random fields, Conditional independence properties , Parameterization of MRFs , Examples , Learning , Conditional random fields (CRFs) , Structural SVMs

Sampling – Basic sampling methods – Monte Carlo Reinforcement Learning, K, Armed Bandit, Elements , Model, Based Learning, Value Iteration, Policy Iteration.

TEXT BOOKS:

1. Christopher Bishop, —Pattern Recognition and Machine Learning| Springer, 2006
2. Kevin P. Murphy, —Machine Learning: A Probabilistic Perspective|, MIT Press, 2012
3. Tom Mitchell, "Machine Learning", McGraw-Hill, 1997.(latest edition)

REFERNCE BOOKS

1. Ethem Alpaydin, —Introduction to Machine Learning|, Prentice Hall of India, 2005
2. Hastie, Tibshirani, Friedman, —The Elements of Statistical Learning| (2nd ed)., Springer, 2008
3. Stephen Marsland, —Machine Learning –An Algorithmic Perspective|, CRC Press, 2009.

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER V	MCA 405	DATA MINING & WAREHOUSING	100	L	T	P	C
				3	1	3	4
GOAL	Understand the fundamentals of data mining and its application in various business and social domains.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Understand the fundamentals of data mining and its functionalities 2. Obtain knowledge in different data mining techniques and algorithms 3. Discuss about various application domains of data mining 4. Understand advanced mining 5. Apply on different case studies			The student should be able to 1. Illustrate the concepts of data mining and data warehousing concepts and techniques. 2. Apply data mining techniques using data mining tools. 3. Implement different data mining techniques and algorithms 4. Do webmining and spatial mining 5. Implement data ware house				

Unit – I

Dss-Uses, definition, Operational Database. Introduction to DATA Warehousing. Data-Mart, Concept of Data-Warehousing, Multi Dimensional Database Structures. Client/Server Computing Model & Data Warehousing. Parallel Processors & Cluster Systems. Distributed DBMS implementations.

Unit – II

DATA Warehousing. Data Warehousing Components. Building a Data Warehouse. Warehouse Database. Mapping the Data Warehouse to a Multiprocessor Architecture. DBMS Schemas for Decision Support. Data Extraction, Cleanup & Transformation Tools. Metadata.

Unit – III

Business Analysis. Reporting & Query Tools & Applications. On line Analytical Processing(OLAP).Patterns & Models. Statistics. Artificial Intelligence.

Unit – IV

Knowledge Discovery, Data Mining. Introduction to Data-Mining. Techniques of Data-Mining. Decision Trees. Neural Networks. Nearest Neighbor & Clustering. Genetic Algorithms. Rule Introduction. Selecting & Using the Right Technique.

Unit – V

Multimedia Data-Mining, Multimedia-Databases, Mining Multimedia Data, Data-Mining and the World Wide Web, Web Data-Mining, Mining and Meta-Data. Data Visualization & Overall Perspective. Data Visualization. Applications of Data-Mining.

References:

1. Berson, “Data Warehousing, Data-Mining & OLAP”, TMH
2. Mallach, “Decision Support and Data Warehousing System”, TMH
3. Bhavani Thura-is-ingham, “Data-Mining Technologies, Techniques Tools & Trends”, CRC Press
4. Navathe, “Fundamental of Database System”, Pearson Education
5. Margaret H. Dunham, “Data-Mining. Introductory & Advanced Topics”, Pearson Education
6. Pieter Adriaans, Dolf Zantinge, “Data-Mining”, Pearson Education

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER VI	Elective – III MCA 406 (i)	Distributed Architecture of Enterprise Applications	100	L	T	P	C
				3	1	3	4
GOAL	To learn the concepts of ERP, Architectures and implementation.						
OBJECTIVES			OUTCOMES				
The course should enable the student to 1. Learn the principles of business engineering. 2. Learn the guidelines for ERP implementation. 3. Learn business modeling and ERP implementation. 4. Learn ERP models and its advantage 5. Learn SAP architectures and its uses			The student should be able to 1. Have an understanding on how business engineering goes with information technology 2. Apply ERP implementation technology 3. Select the ERP models for the application. 4. Have an understanding about the features of SAP package. 5. Develop small applications using SAP.				

UNIT – I

INTRODUCTION TO ERP Integrated Management Information Seamless Integration – Supply Chain Management – Integrated Data Model – Benefits of ERP – Business Engineering and ERP – Definition of Business Engineering – Principle of Business Engineering – Business Engineering with Information Technology.

UNIT – II

BUSINESS MODELLING FOR ERP Building the Business Model – ERP Implementation – An Overview – Role of Consultant, Vendors and Users, Customization – Precautions – ERP Post Implementation Options-ERP Implementation Technology –Guidelines for ERP implementation.

UNIT – III

ERP AND THE COMPETITIVE ADVANTAGE ERP domain MFGPRO – IFS/Avalon – Industrial and Financial Systems – Baan IV SAP-Market Dynamics and Dynamic Strategy.

UNIT – IV

COMMERCIAL ERP Description – Multi-Client Server Solution – Open Technology – User Interface- Application Integration.

UNIT – V

SAP ARCHITECTURE Basic Architectural Concepts – The System Control Interfaces – Services – Presentation Interface – Database Interface.

REFERENCE BOOKS

1. Vinod Kumar Garg and N.K.Venkita Krishnan, —Enterprise Resource Planning – Concepts and Practicell, PHI, 1998.
2. Jose Antonio Fernandz, The SAP R/3 Handbook, TMH, 1998.
3. Bret Wagner, Ellen Monk, —Enterprise Resource Planningll, 3rd Ed., Course Technology, 2008.
4. Alexis Leon, —Enterprise Resource Planningll, 2nd Ed., Tata McGraw Hill, 2007.
5. Mahadeo Jaiswal, Ganesh Vanapalli, —Textbook of Enterprise Resource Planningll, Macmillan Publishers India, 2005.

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER VI	Elective – III MCA 406 (ii)	PARALLEL PROCESSING	100	L	T	P	C
				3	1	3	4
GOAL	To underline the purpose of parallel programming; to learn the techniques that are involved in the design of parallel programming; to develop codes using the tools available for high performance computing; and to know to analyze the performance of the parallel algorithms developed.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Understand the fundamental principles in parallel programming, parallel algorithm design and to survey on some fundamental parallel algorithms 2. Apply the approaches used in high performance computing and program construction with numerous examples to illustrate 3. Undertake an in-depth study of four parallel programming tools: the parallel languages Compositional C++ (CC++), Fortran M (FM), and High Performance Fortran (HPF), and the Message Passing Interface (MPI) library. HPF and MPI are standard parallel programming systems, and CC++ and FM are modern languages particularly well-suited for parallel software engineering. 4. Learn and practice the data transformation and visualization tools for transaction oriented applications			The student should be able to 1. Understand the language design issues related to parallel programming 2. Apply the best practices while designing algorithms for parallel programming 3. Practice the parallel programming using the modern development platforms and tools 4. Apply the concept of parallelism while building the software components or software processes 5. Analyze the performance of the algorithms designed and know to address the issues related				

UNIT-I

DESIGN OF PARALLEL ALGORITHMS Parallel Computers and Computation - A Parallel Machine Model - A Parallel Programming Model - Parallel Algorithm Examples- Partitioning – Communication – Agglomeration – Mapping - Load-Balancing Algorithms- Task-Scheduling Algorithms - Case Studies- Random Numbers Generation -Hypercube Algorithms -Vector Reduction- Matrix Transposition- Mergesort

UNIT-II

APPROACHES TO PERFORMANCE MODELING A Quantitative Basis for Design - Defining Performance- Approaches to Performance Modeling- Developing Models- Performance parameters- time, scalability, overheads, bandwidth, efficiency, speed, - interconnection networks- Input/output- Case Study: Shortest-Path Algorithms-Floyd's Algorithm-Dijkstra's Algorithm -Modular Design Review-Modularity and Parallel Computing-Performance Analysis - Case Study: Convolution, Tuple Space and Matrix Multiplication

UNIT-III

PARALLEL COMPUTING DEVELOPMENT TOOLS -I C++ Review- CC++ Introduction- Concurrency- Locality- Processor Objects- Global Pointers- Thread Placement- Communication- Remote Operations- Synchronization- Mutual Exclusion- Data Transfer Functions- Asynchronous Communication- Determinism- Mapping- Modularity-Performance Issues

UNIT-IV

PARALLEL COMPUTING DEVELOPMENT TOOLS -II Fortran M - Concurrency-Communication- Unstructured Communication- Asynchronous Communication- Determinism-Argument Passing- Mapping- Modularity - High Performance Fortran -Data Parallelism- Concurrency-Data Distribution-Dummy Arguments and Modularity-Other HPF Features-Performance Issues

Unit – V

ADD-ON TOOLS FOR DEVELOPMENT Message Passing Libraries -The MPI Programming Model-MPI Basics - C & Fortran Language bindings with MPI functions- Global Operations-Asynchronous Communication- Modularity-Other MPI Features-Performance Issues-Performance Tools - Performance Analysis- Data Collection-Data Transformation and Visualization-Tools – Paragraph- Upshot—Pablo-Gauge-ParAide- IBM's Parallel Environment-AIMS- Custom Tools

Textbook

1. Ion Foster, Designing and Building Parallel Programs, Addison Wesley, 2003

REFERENCES

1. Arjen Markus, Modern Fortran in Practice, Cambridge University Press, 2012

Online :<http://ebooks.cambridge.org/ebook.jsf?bid=CBO9781139084796>

2. Charles H. Koelbe, High Performance Fortran Handbook, MIT Press 1993

3. Michael J. Quinn, —Parallel Programming in C with MPI and OpenMP, Tata McGraw-Hill Publishing Company Ltd., 2003.

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER VI	Elective – III MCA 406 (iii)	FORENSIC SCIENCE AND APPLICATION	100	L	T	P	C
				3	1	3	4
GOAL	To emphasize the importance of scientific methods in crime detection. To provide a platform for students and forensic scientists to exchange views, chalk out collaborative programs and work in a holistic manner for the advancement of forensic science.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. The significance of forensic science to human society. 2. The fundamental principles and functions of forensic science 3. The divisions in a forensic science laboratory. 4. The working of the forensic establishments in India and abroad. 5. The fundamental principles on which the science of fingerprinting is based. 5. Gain knowledge for instrumentation and investigation techniques.			The student should be able to 1. The significance of forensic science to human society. 2. The fundamental principles and functions of forensic science 3. The importance of chromatographic and spectroscopic techniques in processing crime scene evidence. 4. The methods of securing, searching and documenting crime scenes 5. The method of classifying criminal record by fingerprints was worked out in India, and by Indians. 6. The significance of microscopy in visualizing trace evidence and comparing it with control samples 7. The classification of firearms and their firing mechanisms. The methods of identifying firearms.				

UNIT I:

Forensic Science - History and development of Forensic Science Historical aspects of forensic science, Definitions and concepts of forensic science, Need of Forensic Science, Basic Principles of Forensic Science, Scope of development of forensic science. Functions of Forensic Science, Different branches of Forensic Science. Frye case and Daubert standard. Scope and development of forensic science.

UNIT II:

Legal aspects of crime Crime – Introduction Natures, causes and consequences of crime, Broad concepts of criminal Justice system, Procedures involved in the detection of crime, Filing of criminal charges, Indian police system – The Police Act, Human rights and criminal justice system in India. Set up of INTERPOL. Duties and qualification of forensic science.

UNIT III:

Organizational set up of FSL in India Hierarchical set up of central forensic science laboratory, Hierarchical set up state forensic science laboratory, Government examiners of questioned documents. Chemical examiners laboratory, Finger print bureaus, National crime records bureau, Bureau of police research and development, Mobile crime laboratory, Duties of forensic scientist, code of conduct of forensic scientists. Drug enforcement administrator. Defense research and development organization.

UNIT IV:

Administration and Organizational Setup: DFSS, CFSL, GEQD, SFSL, RFSL, MFSL, FPB, NICFS, CDTS, NCRB, BPR&D, Qualifications and duties of Forensic Scientists

Police and Forensic Science: Relationship between police and forensic expert, Role of Police at the Crime scene, scientific help at crime scene, handling of various types of crime scenes by police, forensic teaching of police personals, forensic case documentation by Police, Technological Advance and Police.

UNIT V: Instrumentation & Investigation Techniques

Crime Detection Devices: UV, IR, X-Rays, their nature and applications, Detective Dyes, Neutron Radiography, Speed Detection Devices

Tools: Basic Kits, Investigator's Kit, Tools used in Mobile laboratory.

Microscopy: Definition, Types of Microscopes and their Forensic Significance

Chromatography: Definition, Types of Chromatography and Forensic Applications with reference to TLC, Preparation of TLC Plate.

Documents: Definition, Importance, Nature, Problems and preliminary examination.

Ballistics: Definition of Ballistics, Forensic Ballistics, Types of Forensic Ballistics, Firearms, History of Firearm, Classification of Modern Firearms, Ammunition and its composition, GSR and its examination.

Fingerprints Examination: History & Development, Fundamental Principles, Types of Fingerprints, Classification of Fingerprints, Fingerprint characteristics, Recent Advanced Techniques used in Forensic Science : Lie Detection, Voice Identification, DNA Profiling, Narco Analysis, Brain Fingerprinting.

Text Books:

1. B.B. Nanda and R.K Tiwari, Forensic Science in India : A vision for the Twenty First Century, select publishers, New Delhi(2001)
2. M.K Bhasin and S.Nath, Role of Forensic Science in the New Millenium, University of Delhi, Delhi(2002)
3. S.H James and J.J Nordby, Forensic Science :An introduction to scientific and Investigative Techniques, 2nd Edition, CRC Press, Boca Raton(2005)
4. W.G. Eckert and R.K. Wright in Introduction to Forensic Sciencies, 2nd Edition, W.G. Eckert (ED), CRC Press, Boca Raton(1997)
5. R. Saferstein, M.L. Hastrup and C.Hald, Fisher's Techniques of Crime scene Investigation, CRC Press, Boca Raton (2013)
6. W.J. Tilstone, M.L. Hastrup and C.Hald, Fisher's Techniques of Crime Scene Investigation, CRC Press, Boca Raton (2013)

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER VI	Elective – III MCA 406 (iv)	DIGITAL IMAGE PROCESSING	100	L	T	P	C
				3	1	3	4
GOAL	To improve the visual effect of people for human beings. The main purpose of DIP is Visualization, Image sharpening and restoration and Image Variation.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Gain knowledge in Digital image. 2. Understand the concepts and applications of Digital image Processing. 3. It helps to improve images for human interpretation. 4. Information can be processed and extracted from images for machine interpretation. 5. Image Processing extracts information from images and integrates it for several applications.			The student should be able to 1. Understanding of Digital image Processing. 2. The ability to perform filtering, interpolation & Transformation Tools. 3. The ability to perform Image compression & Applications 4. The ability to perform Image Transforms & Applications				

UNIT-I:

Introduction: Fundamental steps of image processing, components of an image processing of system, the image model and image acquisition, sampling and quantization, station ship betweenpixels, distance functions, scanner

UNIT-II:

Statistical and spatial operations: Grey level transformations, histogram equalization, smoothing & sharpening-spatial filters, frequency domain filters, homomorphic filtering, image filtering & restoration. Inverse and weiner filtering. FIR weiner filter, Filtering using image transforms, smoothing splines and interpolation.

UNIT-III:

Morphological and other area operations: basic morphological operations, opening and closing operations, dilation erosion, Hit or Miss transform, morphological algorithms, extension to grey scale images. Segmentation and Edge detection region operations, basic edge detection, second order detection, crack edge detection, gradient operators, compass and laplace operators, edge linking and boundary detection, thresholding, region based segmentation, segmentation by morphological watersheds.

UNIT-IV:

Image compression: Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression, image data compression-predictive technique, pixel coding, transfer coding theory, lossy and lossless predictive type coding. Basics of color image processing, pseudocolor image processing, color transformation, color smoothing and sharpening, color segmentation, color image compression, compression standards

UNIT-V:

Image Transforms - Fourier, DFT, DCT, DST, Haar, Hotelling, Karhunen -Loeve, Walsh, Hadamard, Slant. Representation and Description - Chain codes, Polygonal approximation, Signatures Boundary Segments, Skeltons, Boundary Descriptors, Regional Descriptors, Relational Descriptors, PCA.

REFERENCES:

1. Digital Image Processing – by Rafael.C.Gonzalez & Richard E.Woods, 3rd edition, Pearson Education, 2008.
2. Digital Image Processing, M.Anji Reddy, Y.Hari Shankar, BS Publications.
3. Fundamentals of Digital Image Processing – by A.K. Jain, PHI.
4. Digital Image Processing – William K, Part I - John Wiley edition.
5. Digital Image Processing using MATLAB – by Rafael.C.Gonzalez, Richard E.Woods, & Steven L.Eddins, Pearson Education, 2006
6. Digital Image Processing, Kenneth R. Castleman, Pearson Education, 2007.

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER VI	Elective – III MCA 406 (v)	CLOUD COMPUTING	100	L	T	P	C
				3	1	3	4
GOAL	Understand the architecture of Cloud and industry frameworks.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. To study about migrating into cloud. 2. To study IAS and enterprise cloud. 3. To study security aspects in cloud and integration of cloud. 4. To study the security aspects of the cloud 5. Understand inter cloud environment			The student should be able to 1. Understand migration. 2. Explain issues for enterprise application. 3. Understand cloud security and integration of cloud. 4. Understand the security aspects of the cloud 5. To understand issues in inter cloud environment like load balancing, optimization and reconfiguration				

UNIT I

CLOUD COMPUTING Introduction to Cloud Computing, Definition, Characteristics, Components, Cloud provider, SAAS, PAAS, IAAS and Others, Organizational scenarios of clouds, Administering & Monitoring cloud services, benefits and limitations, Deploy application over cloud, Comparison among SAAS, PAAS, IAAS Cloud computing platforms: Infrastructure as service: Amazon EC2, Platform as Service: Google App Engine, Microsoft Azure, Utility Computing, Elastic Computing.

UNIT II

CLOUD TECHNOLOGY Introduction to Cloud Technologies, Study of Hypervisors Compare SOAP and REST Webservices, AJAX and mashups-Web services: SOAP and REST, SOAP versus REST, AJAX: asynchronous 'rich' interfaces, Mashups: user interface services Virtualization Technology: Virtual machine technology, virtualization applications in enterprises, Pitfalls of virtualization. Multitenant software: Multi-entity support, Multi-schema approach, Multi-tenance using cloud data stores, Data access control for enterprise applications.

UNIT III

DATA IN THE CLOUD Data in the cloud: Relational databases, Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo. Map-Reduce and extensions: Parallel computing, The map-Reduce model, Parallel efficiency of Map-Reduce, Relational operations using Map-Reduce, Enterprise batch processing using Map-Reduce, Introduction to cloud development, Example/Application of Mapreduce, Features and comparisons among GFS,HDFS etc, Map-Reduce model.

UNIT IV

CLOUD SECURITY Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud. Cloud computing security architecture: Architectural Considerations- General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro-architectures; Identity Management and Access control-Identity management, Access control, Autonomic Security. Cloud computing security challenges: Virtualization security management-virtual threats, VM Security Recommendations, VM-Specific Security techniques, Secure Execution Environments and Communications in cloud.

UNIT V

INTERCLOUD ENVIRONMENTS Issues in cloud computing, Implementing real time application over cloud platform. Issues in Intercloud environments, QOS Issues in Cloud, Dependability, data migration, streaming in Cloud. Quality of Service (QoS) monitoring in a Cloud computing environment. Cloud Middleware. Mobile Cloud Computing. Inter Cloud issues. A grid of clouds, Sky computing, load balancing, resource optimization, resource dynamic reconfiguration, Monitoring in Cloud.

TEXT BOOKS

1. Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper,—Cloud Computing for Dummies®, Wiley, 2009.
2. Gautam Shroff, —Enterprise Cloud Computing®, Cambridge, 2010.
3. Ronald Krutz and Russell Dean Vines, —Cloud Security: A Comprehensive Guide to Secure Cloud Computing®, Wiley, 2010.

REFERENCE BOOKS

1. Scott Granneman,—Google AppsDeciphered: Compute in the Cloud to Streamline Your Desktop®, Pearson Education, 2008.
2. Tim Malhar, S.Kumaraswammy, S.Latif, —Cloud Security & Privacy®, O'Reilly, 2009.
3. Toby Velte, Anthony Velte, Robert Elsenpeter,—Cloud Computing : A Practical Approach®, McGraw Hill, 2009.
4. Barrie Sosinsky, —Cloud Computing Bible®, Wiley, 2011.

PAPER NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
PAPER VI	Elective – III MCA 406 (vi)	GRID COMPUTING	100	L	T	P	C
				3	1	3	4
GOAL	To impart knowledge on genesis and applications of Grid technology.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Understand the genesis of grid computing 2. Know the application of grid computing 3. Understand the technology and tool kits for facilitating grid computing 4. Understand OGSA services 5. Understand Globus toolkit			The student should be able to 1. Have the ability to understand the Grid services. 2. Acquire knowledge on open Grid service infrastructure. 3. Program using tool kit. 4. Acquire knowledge on OGSA services 5. Acquire knowledge on globus toolkit				

UNIT I

INTRODUCTION GRID COMPUTING: Introduction: Early Grid Activities, Current Grid Activities, An Overview of Grid business Areas, Grid Applications. Grid Computing Organizations and Their Roles: Organizations Developing Grid Standards and Best Practice Guidelines, Organizations Developing Grid Computing Toolkits and the Framework. The Grid computing Anatomy: The Grid Problem. The Grid Computing Road Map.

UNIT II

MESSAGING Merging the Grid Services Architecture with the Web Services Architecture: Service-Oriented Architecture, Web Service Architecture, XML, Related Technologies, and Their Relevance to Web Services, XML Messages and Enveloping, Service Message Description Mechanisms, Relationship between Web Service and Grid Service.

UNIT III

OPEN GRID SERVICES Open Grid Services Architecture (OGSA): Some Sample Use cases that drive the OGSA, CDC, NFS, Online Media and Entertainment. OGSA Platform Components. Open Grid Services Infrastructure (OGSI): Introduction, Grid Services, High-Level Introduction to OGSI, Technical Details of OGSI specification, Introduction to Service Data Concepts, Grid Service : Naming and Change Management Recommendations.

UNIT IV

OGSA BASIC SERVICES OGSA Basic Services: Common Management Model (CMM), Service domains, Policy Architecture, Security Architecture, Metering and Accounting, Common distributed Logging, Distributed Data Access and Replication.

UNIT V

GLOBUS TOOLKIT GLOBUS TOOLKIT: Architecture: GT3 software Architecture Model. GLOBUS TOOLKIT: Programming Model - Introduction, Service Programming Model. GLOBUS TOOLKIT: A Sample Implementation, Acme Search Service Implementation in a Top-down Approach.

TEXT BOOK

1. Joshy Joseph and Craig Fellenstein, —Grid Computing], Pearson Education, 2003.

REFERENCE BOOKS

2. Fran Berman, Geoffrey Fox, Anthony J.G. Hey, —Grid Computing: Making the Global Infrastructure a Reality —, John Wiley and Sons, 2003.
3. Ahmar Abbas, —Grid Computing: A Practical Guide to Technology and Applications—, Charles River Media, 2003.
4. D Janaki Ram, —Grid Computing—, TMH.

LAB NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
LAB I	MCA 407	BIG DATA LAB	50	L	T	P	C
				0	0	3	2
GOAL	To increase the speed at which products get to market, to reduce the amount of time and resources required to gain market adoption, target audiences, and to ensure customers remain satisfied.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Gain knowledge in Big Data. 2. Understand the concepts and applications of Big Data. 3. An organisation or individual can obtain, store, transform and analyse large amounts of data to solve specific problems. 4. A data driven approach to understanding a business. One can build predictive data models and detect future trends. Devices of the future will be built entirely on big data.			The student should be able to 1. Understanding of Big Data 2. Implement various NoSQL Data Management 3. Develop applications using Hadoop 4. Implement Map Reduce for different applications. 5. Implement Hadoop related tools 6. Using big data helps you increase sales and loyalty, increases your efficiency, improves your pricing & ensures you hire the right employees.				

1. Implement the following Data structures in Java
 - a) Linked Lists b) Stacks c) Queues d) Set e) Map
2. (i) Perform setting up and Installing Hadoop in its three operating modes:
 - Standalone,
 - Pseudo distributed,
 - Fully distributed

(ii) Use web-based tools to monitor your Hadoop setup.

3. Implement the following file management tasks in Hadoop:
 - Adding files and directories
 - Retrieving files
 - Deleting files

Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.

4. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

5. Write a Map Reduce program that mines weather data.

Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with Map Reduce, since it is semi structured and record-oriented.

6. Implement Matrix Multiplication with Hadoop Map Reduce

7. Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.

8. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes

9. Creating the HDFS tables and loading them in Hive and learn joining of tables in Hive

10. Basic CRUD operations in Apache Cassandra

11. Develop a system which can use of Web search, web crawlers and web information retrieval to store the Data into Cassandra Database.

12. Analyze and implement a system with Web graph mining.

13. Implement and Subscribe RSS News feeds to get latest news in India.

- 14 Installation of Apache Spark, Exploring RDDs using Spark Shell, Process Data Files with Apache Spark, Use Pair RDDs to Join Two Datasets
15. Write and Run an Apache Spark Application, Configure an Apache Spark Application
16. View Jobs and Stages in the Spark Application UI, Persist an RDD and Implement an Iterative Algorithm with Apache Spark
17. Use Apache Spark SQL for ETL
18. Write an Apache Spark Streaming Application
19. Process Multiple Batches with Apache Spark Streaming

LAB NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
LAB II	MCA 408	PYTHON PROGRAMMING LAB	50	L	T	P	C
				0	0	3	2
GOAL	To emphasizes programmer productivity and code readability. To build websites and software, automate tasks, and conduct data analysis.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Gain knowledge in Python programming. 2. Understand the concepts and applications of Python programming. 3. Build basic programs using fundamental programming constructs like variables, conditional logic, looping, and functions. 4. To provide problem solving and programming capability.			The student should be able to 1. Understanding of Python programming. 2. Build basic programs using conditional logic, looping, List and functions 3. Implement Database Access, GUI Programming for different applications. 4. Implement Networking, XML Processing, for different applications. 5. Develop applications using Python programming.				

1. Python Basic: Python Data Types - String , List , Dictionary, Tuple, Sets
2. Python Array
3. Python Conditional statements and loops
4. Python functions
5. Python Data Structures and Algorithms
6. Data Structure
7. Search and Sorting
8. Recursion
9. Python Date Time
10. Python Class
11. Python Math
12. Python File Input Output
13. Python Regular Expression
14. Python NumPy :Python NumPy arrays, Python NumPy Random, Python NumPy Math, Statistics, Trigonometry and Linear algebra , Python NumPy DateTime
15. Python Pandas : Python Pandas - DataFrame and DataFrame
16. Python Web Scraping : Python Web Scraping

LAB NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
LAB III	MCA 409	DATA SCIENCE LAB	100	L	T	P	C
				3	1	3	4
GOAL	To extract, pre-process and analyze data and help solve problems. To make businesses grow better.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Gain knowledge in Data science. 2. Understand the concepts and applications of Data science. 3. To explore, sort and analyze mega data from various sources in order to take advantage of them and reach conclusions to optimize business processes or for decision support. 4. To construct the means for extracting business-focused insights from data.			The student should be able to 1. Understanding of data science. 2. Build Python programs for data science using functions, OOPs and file handling and numpy 3. Using Pandas Implement Data manipulation and Database for different applications. 4. Implement data visualization with MATPLOTLIB for different applications. 5. Develop Data Science applications using Python programming and Machine Learning				

1. Interactive commands in Python, data operations, simple programs for writing into files and reading from files.
2. Data file manipulations programs.
3. Familiarization with IDE (PyCharm /Spyder) in Python.
4. Writing programs for standard algorithms of sorting and searching in Python.
5. Introduction to Python Libraries- NumPy, Pandas, Matplotlib, Scikit
6. Perform Data exploration and preprocessing in Python
7. Perform CRUD operation using SQL
8. Perform Joining Multiple Tables Data using difference type of SQL Joins
9. Perform Data exploration and preprocessing Using SQL
10. Plotting the data using X-Y graph, Bar- chart, and using other plotting techniques.
11. Write programs to perform exploratory data analysis: variance, standard derivation, summarization, distribution, and statistical inference.
12. Plotting the various distributions for given data sets.
13. Implement Linear regression
14. Implement logistic regression
15. Implement Naive Bayes classifier for dataset stored as CSV file.
16. Build models using Decision trees
17. Build model using SVM with different kernels
18. Classifying and presentation of data using support vector machine.
19. Implement K-NN algorithm to classify a dataset.
20. Build models using different Ensembling techniques
21. Writing Flask API for models
22. Deployment of models on clouds(AWS/Azure/GCP)

LAB NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
LAB IV	MCA 410	MACHINE LEARNING LAB	50	L	T	P	C
				0	0	3	2
GOAL	To impart knowledge on Machine Learning						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. To understand the concepts of machine learning 2. To learn supervised and unsupervised learning and their applications 3. To understand the theoretical and practical aspects of Probabilistic Graphical Models 4. To appreciate the concepts and algorithms of reinforcement learning 5. To learn aspects of computational learning theory			The student should be able to 1. Upon Completion of the course, the students will be able to 2. Apply neural networks for suitable application. 3. Implement probabilistic discriminative and generative algorithms for an application of your choice and analyze the results 4. Use a tool to implement typical clustering algorithms for different types of applications 5. Design and implement an HMM for a sequence model type of application				

1. Study and Implement the Naive Bayes learner using Sklearn. (The datasets taken can be: Breast Cancer data file or Reuters data set).
2. Study and Implement the Decision Tree learners using Sklearn. (The datasets taken can be: Breast Cancer data file or Reuter's data set).
3. Estimate the accuracy of decision classifier on breast cancer dataset using 5-fold cross-validation. (You need to choose the appropriate options for missing values).
4. Estimate the precision, recall, accuracy, and F-measure of the decision tree classifier on the text classification task for each of the 10 categories using 10-fold cross-validation.
5. Write a program to develop a machine learning method to classifying your incoming mail using Sklearn.
6. Write a program to develop a machine learning method to Predict stock prices based on past price variation using Sklearn.
7. Write a program to develop a machine learning method to predict how people would rate movies, books, etc. using Sklearn.
8. Write a program to develop a machine learning method to Cluster gene expression data, how to modify existing methods to solve the problem better
9. Select two datasets. Each dataset should contain examples from multiple classes. For training purposes assume that the class label of each example is unknown (if it is known, ignore it). Implement the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameter k.
10. Suggest and test a method for automatically determining the number of clusters.
11. Using a dataset with known class labels compare the labeling error of the K-means and EM algorithms. Measure the error by assigning a class label to each example. Assume that the number of clusters is known.
12. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
13. Build an Artificial Neural Network by implementing CNN algorithm and test the same using appropriate data sets.

LAB NO.	COURSE CODE	COURSE TITLE	MM	PERIODS			
LAB V	MCA 411	DATA MINING & WAREHOUSING LAB	50	L	T	P	C
				0	0	3	2
GOAL	To understand and demonstrate the basic concepts of data mining using open sourece tools.						
OBJECTIVES			OUTCOMES				
The course should enable the students to 1. Learn a data mining tool. 2. Apply data mining tools for various phases of data mining. 3. Apply the OLAP, rollup, drill down operation 4. Understand about generalization and summarization technique 5. Apply association rule for mining 6. Apply the clustering technique 7. Apply Naïve Bayes classifier. 8. Design a decision tree.			The student should be able to 1. Explain the features of selected tool. 2. Implement various data mining techniques using the selected tool. 3. Implement the OLAP, rollup, drill down operation. 4. Implement generalization and summarization technique 5. Implement association rule for mining 6. Implement the clustering technique 7. Implement Naïve Bayes classifier. 8. Construct a decision tree.				

LIST OF EXPERIMENTS-1

1. Evolution of data management technologies, introduction to data warehousing concepts.
2. Develop an application to implement defining subject area, design of fact dimensiontable, data mart.
3. Develop an application to implement OLAP, roll up, drill down, slice and dice operation
4. Develop an application to construct a multidimensional data.
5. Develop an application to implement data generalization and summarization technique.
6. Develop an application to extract association rule of data mining.
7. Develop an application for classification of data.
8. Develop an application for one clustering technique
9. Develop an application for Naïve Bayes classifier.
10. Develop an application for decision tree.

LIST OF EXPERIMENTS-1

1. Create an Employee Table with the help of Data Mining Tool WEKA.
2. Create a Weather Table with the help of Data Mining Tool WEKA.
3. Apply Pre-Processing techniques to the training data set of Weather Table
4. Apply Pre-Processing techniques to the training data set of Employee Table
5. Normalize Weather Table data using Knowledge Flow.
6. Normalize Employee Table data using Knowledge Flow.
7. Finding Association Rules for Buying data.
8. Finding Association Rules for Banking data

9. Finding Association Rules for Employee data.
10. To Construct Decision Tree for Weather data and classify it.
11. To Construct Decision Tree for Customer data and classify it.
12. To Construct Decision Tree for Location data and classify it.
13. Write a procedure for Visualization for Weather Table.
14. Write a procedure for Visualization of Banking Table.
15. Write a procedure for cross-validation using J48 Algorithm for weather table
16. Write a procedure for Clustering Buying data using Cobweb Algorithm
17. Write a procedure for Clustering Weather data using EM Algorithm.
18. Write a procedure for Banking data using Farthest First Algorithm.
19. Write a procedure for Employee data using Make Density Based Cluster Algorithm.
20. Write a procedure for Clustering Customer data using Simple KMeans Algorithm
21. List all the categorical (or nominal) attributes and the real-valued attributes separately.
22. What attributes do you think might be crucial in making the credit assessment? Come up with some simple rules in plain English using your selected attributes.
23. *What attributes do you think might be crucial in making the bank assessment?
24. One type of model that you can create is a Decision Tree -train a Decision Tree using the complete dataset as the training data. Report the model obtained after training.
25. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy?
26. *Find out the correctly classified instances, root mean squared error, kappa statistics, and mean absolute error for weather data set?