SCHEME OF COURSES FOR MCA(2015-18)

SEMESTER-I

S.No.	Course No.	Title	L	T	P	Cr
1	MCA101	Discrete Mathematics	3	1	0	3.5
2	MCA102	Computer Programming	3	0	4	5.0
3	MCA103	Computer Organization and Architecture	3	1	2	4.5
4	MCA107	Statistical and Numerical Methods	3	1	2	4.5
5	MCA104	LAMP and Web Designing	2	0	4	4.0
		Total	14	3	12	21.5

SEMESTER-II

S.No.	Course No.	Title	L	T	P	Cr
1	MCA207	Personality Development and Communication Skills	3	0	2	4.0
2	MCA202	Graph Theory and its Applications	3	1	0	3.5
3	MCA203	Object oriented Programming	3	1	2	4.5
4	MCA204	Programming Languages	3	0	2	4.0
5	MCA205	Engineering Design Project (4 self effort Hours)	1	0	4	5.0
		Total	13	2	10	21

SEMESTER-III

S.No.	Course No.	Title	L	T	P	Cr
1	MCA301	Computer Networks	3	0	2	4.0
2	MCA302	Operating Systems	3	0	2	4.0
3	MCA303	Information Management Systems	3	0	4	5.0
4	MCA304	Data Structure and Algorithms	3	0	4	5.0
5	MCA305	Software Engineering and Project Management	3	0	2	4.0
		Total	15	0	14	22

SEMESTER-IV

S.No.	Course No.	Title	L	T	P	Cr
1	MCA401	Automata Theory and Computability	3	1	0	3.5
2	MCA402	Graphics and Visual Computing	3	0	2	4.0
3	MCA403	Mobile Application Development	3	0	2	4.0
4	MCA404	Big Data Analytics and Business Intelligence	3	1	2	4.5
5	MCA405	Artificial Intelligence	3	0	2	4.0
		Total	15	3	8	20

SEMESTER-V

S.No.	Course No.	Title	L	T	P	Cr
1		Elective-I	3	0	2	4.0
2		Elective-II	3	0	2	4.0
3		Elective-III	3	0	2	4.0
5	MCA505	Capstone project (4 Self Effort Hours)	1	0	6	6.0
		Total	13	0	12	18

SEMESTER-VI (CGPA below 6.0 or Emoluments less than 10000 PM #)

S.No.	Course No.	Title	L	T	P	Cr
1	MCA601	Capstone Project-II (Self-Effort8 Hours)	0	0	4	6.0
2	MCA602	Advanced Application Development	2	0	2	3.0
3	MCA603	Advance Java Programming	2	0	4	4.0
4	MCA604	Enterprise Resource Planning	2	0	2	3.0
		Total	6	0	10	16

Conditions apply

S.No.	Course No.	Title	L	T	P	Cr
1	MCA606	Project Semester	-	-	-	16.0
		Total	-	-	-	16.0
		Grand Total of All Semesters				**128.5**

LIST OF ELECTIVES (I, II, III, IV)

S.No.	Course No.	Title	L	T	P	Cr
1.	MCA501	Network Security	3	0	2	4.0
2.	MCA502	Parallel and Distributed Computing	3	0	2	4.0
3.	MCA503	Human Computer Interaction	3	0	2	4.0
4.	MCA504	Web Analytics and Intelligence	3	0	2	4.0
5.	MCA505	Database Administration	3	0	2	4.0
6.	MCA506	Cloud Infrastructure and Services	3	0	2	4.0
7.	MCA507	User Centred Design	3	0	2	4.0
8.	MCA508	Image and Video Processing	3	0	2	4.0
9.	MCA509	Compiler Construction	3	0	2	4.0
10.	MCA510	Agile Software Development	3	0	2	4.0

MCA101 DISCRETE MATHEMATICS

L	T	P	Cr
3	1	0	3.5

Course Objective: Detailed study of various discrete and algebraic structures, basic logic, basics of counting and proof techniques.

Sets, Relations, and Functions: Sets: Operations on set, Representation of Discrete Structures, Fuzzy set, Multi-set, bijective function, Inverse and Composition of functions, Floor and Ceiling functions, Growth of functions: Big-O notation, Big-Omega and Big-Theta Notations, Determining complexity of a program, Hashing functions, Recursive function, Functions applications.

Relations: Reflexivity, symmetry, transitivity, Equivalence and partial-ordered relations, Asymmetric, Irreflexive relation, Inverse and complementary relations, Partition and Covering of a set, N-ary relations and database, Representation relation using matrices and digraph, Closure of relations, Warshall's algorithm, Lexicographic ordering, Hasse diagram, Lattices, Boolean algebra, Representation and Minimizations of Boolean functions, Application of transitive closure in medicine and engineering. Application: Embedding a partial order.

Basic Logic: Propositional logic, Logical connectives, Truth tables, Normal forms (conjunctive and disjunctive), Validity of well-formed formula, Propositional inference rules (concepts of modus ponens and modus tollens), Predicate logic, Universal and existential quantification, Limitations of propositional and predicate logic.

Proof Techniques: Notions of implication, equivalence, converse, inverse, contra positive, negation, and contradiction, The structure of mathematical proofs, Direct proofs, Disproving by counter example, Proof by contradiction, Induction over natural numbers, Structural induction, Weak and strong induction (i.e., First and Second Principle of Induction), Recursive mathematical definitions, Well orderings.

Algebraic Structures: Group, Semi group, Monoids, Homomorphism, Congruencies, Ring, Field, Homomorphism, Congruencies, cosets and Lagrange Theorem, Group Codes, Integral domain, coding theory, Polynomial rings and polynomial codes, Applications of algebra to control structure of a program, The application of Residue Arithmetic to Computers.

Basics of Counting: Counting arguments: Set cardinality and counting, Sum and product rule, Inclusion-exclusion principle, Arithmetic and geometric progressions, The pigeonhole principle, Permutations and combinations, Solving recurrence relations, Basic modular arithmetic.

Recommended Books:

1. Rosen, K.H., Discrete Mathematics and its Applications, McGraw Hill (2011), 7th ed.
2. Tremblay, J.P. and Manohar R., "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill (2007), 1st ed.
3. Haggard G., Schlipf J. and Whitesides, Sue, Discrete Mathematics for Computer Science, Cengage Learning , (2008), 2nd ed.
4. Johnsonbaugh R., Discrete Mathematics, Pearson Education ,(2007), 7th ed.

COURSE OUTCOMES (COs)

CO1	To understand some fundamental mathematical concepts and terminology.
CO2	To learn and analyze recursive definitions and different types of discrete structures.
CO3	To get familiar with the techniques for constructing mathematical proofs.
CO4	To learn theorem of Lagrange for groups, subgroups, cosets of subgroups, the order of an element.
CO5	To know concepts from the theory of rings such as zero divisor, division rings and fields.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA102 COMPUTER PROGRAMMING

L	T	P	Cr
3	0	4	5.0

Course objective: This course is designed to explore computing and to show students the art of computer programming. Students will learn some of the design principles for writing good programs.

Introduction to 'C' programming: Fundamentals, Structure of a 'C' program, Compilation and linking processes.

Expressions and Console I/O : Basic Data types, Identifier Names, Variables, Scope, Type qualifiers, Storage class specifiers, Constants, Operators, Reading and writing characters, Reading and writing strings, Formatted and console I/O, printf(), scanf(), Suppressing input.

Statements: True and False in C, Selection statements, Iteration statements, Jump statements, Expression statements, Block statements.

Arrays and Strings: Single dimension array, Two dimension array, Strings, Array of strings, Multi-dimension array, Array initialisation, Variable length arrays.

Pointers: Pointer variables, Pointer operators, Pointer expressions, Pointers and arrays, Multiple indirection, Pointer initialization, Pointers to arrays, Dynamically allocated arrays, Problems with pointers.

Functions: General form of a function, Understanding scope of a function, Function arguments, Command line arguments, Return statement, Recursion, Function prototype, Pointers to functions.

Structures, Unions, Enumerations, and Typedef: Structures, Array of structures, Passing structures to functions, Structure pointers, Arrays and structures within structures, Unions, Bit-fields, Enumerations, typedef.

File I/O: Streams and files, File system basics, fread() and fwrite(), fseek() and random access I/O, fprintf() and fscanf(), Standard streams.

Pre-processor and Comments: Pre-processor, #define, #error, #include, Conditional compilation directives, #undef, Single line and multiple line comments.

Laboratory work: To implement Programs for various kinds of programming constructs in C Language.

Recommended Books:

1. Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, Prentice Hall(1988), 2nd ed.
2. Schildt H., *C:* The Complete Reference, Tata Mcgraw Hill, (2000), 4thed.
3. Stroustrup B., The C++ Programming Language, Addison Wesley (2000), 3rd ed.
4. Kanetkar Y., Let Us C, BPB Publications(2012),13th ed.

COURSE OUTCOMES (COs)

CO1	Learn the implementation of simple 'C' program, data types and operators and Console I/O function.
CO2	Learn the Implementation of decision control statements, loop control statements and case control structures.
CO3	Understand the declaration and implementation of arrays, pointers and functions.
CO4	Learn the structures declaration, initialization and implementation.
CO5	Understand the file operations, Character I/O, String I/O, File pointers and importance of pre-processor directives

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA103 COMPUTER ORGANIZATION AND ARCHITECTURE

L	T	P	Cr
3	1	2	4.5

Course Objective: Focus is on the architecture and organization of the basic computer modules viz control unit, central processing unit, input-output organization and memory unit. Covers basics of computer arithmetic and parallel processing concepts.

Basics of Digital Electronics: Codes, Logic gates, Flip flops, Registers, Counters, Multiplexer, Demultiplexer, Decoder, Encoder.

Register Transfer and Micro operations: Register transfer Language, Register transfer, Bus & memory transfer, Logic micro operations, Shift micro operation.

Basic Computer Organization: Instruction codes, Computer instructions, Timing & control, Instruction Cycles, Memory reference instruction, Input/Output& Interrupts, Complete computer description & design of basic computer.

Control Unit: Hardwired vs. Micro programmed control unit.

Central Processing Unit: General register organization, Stack organization, Instruction format, Data transfer & manipulation, Program control, RISC, CISC.

Computer Arithmetic: Addition & subtraction, Multiplication Algorithms, Division algorithms.

Input-Output Organization: Peripheral devices, I/O interface, Data transfer schemes, Program control, Interrupt, DMA transfer, I/O processor.

Memory Unit: Memory hierarchy, Processor vs. memory speed, High-speed memories, Cache memory, Associative memory, Interleave, Virtual memory, Memory management.

Introduction to Parallel Processing: Pipelining, Characteristics of multiprocessors, Interconnection structures, Interprocessor arbitration, Interprocessor communication & synchronization.

Case Studies: Case studies of some contemporary advanced architecture for processors of families like Intel, AMD, IBM etc./Seminar on State-of the-art technology.

Lab Work : To implement different programs using ARM processor

Recommended Books

1.Mano, Morris M., Computer System Architectue, Prentice Hall (1992) , 3rd ed.

2.Hayes, J.P., Computer Architecture and Organization, McGraw Hill (1998), 3rded.

3.Hennessy, J.L., Patterson, D.A, and Goldberg, D., Computer Architecture A Quantitative Approach, Pearson Education Asia (2006), 5thed.

4.Leigh, W.E. and Ali, D.L., System Architecture: software and hardware concepts, South Wester Publishing Co. (2000).

COURSE OUTCOMES (COs)

CO1	Understand the working of logic gates, Flip flops, Registers, Counters, Multiplexer, Demultiplexer, Decoder, Encoder etc.
CO2	Learn the syntax of Register transfer Language and different micro operations
CO3	Design and analyze the instruction format & addressing modes for a given operation and algorithms for addition, subtraction, multiplication & division.
CO4	Understand the working of different types of memories like associate memory, cache memory, virtual memory etc. and Interfaces of Computer with input and output devices
CO5	Understand the concept of pipelining, multiprocessors, and inter processor communication and comparison of different contemporary advanced architecture for processors of families like Intel, AMD, IBM.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA104 LAMP AND WEB DESIGNING

L	T	P	Cr
2	0	4	4.0

Course objective: To give a brief introduction to the open source technology. Through interactive sessions enabling students to enhance their skills in contributing and implementing their technical knowledge.

Introduction: Open Source Definition, Free Software vs. Open Source Software, Public Domain Software, Open Source History, Initiatives, Principle and Methodologies. Open Standards.

Open Source Development Model Licenses and Patents: What Is a License, Important FOSS Licenses (Apache, BSD, GPL, LGPL), copyrights and copy lefts, Patents Economics of FOSS: Zero Marginal Cost, Income-generation opportunities, Problems with traditional commercial software, Internationalization.

Open Source Operating Systems: Different open source operating systems. Google Chrome OS, BSD, Linux Distributions – Fedora and Ubuntu, Installation, Disk Partitioning, Boot loader. Using Linux – Shell, File system familiarity, Linux Administration – Managing users, services and software, Network Connectivity, Configurations and Security.

Open Source Web Technologies: Two Tier and Three Tier Web based Application Architecture. LAMP Terminologies, Advantages . Apache, Web server conceptual working, Web browser, HTTP, Installation and Configuration, httpd.conf file, Logging, Security, Running a website, MySQL, Database management system, ER diagram, Relational database, Installation, Configuration, Administration, Common SQL queries.PHP**,** Dynamic content, Server side scripting, Installation, Configuration, Administration, Language syntax, Built-in functions, PHP and MySQL connectivity.

Programming on XHTML and XML: Editing XHTML, W3C XHTML validation services, designing XHTML by using XHTML tables, frames, forms and other elements. CSS and its types. XML, XML namespaces, DTD, XML schema, XML vocabularies, DOM and its methods, SOAP.

Programming on PHP and JavaScript:

JavaScript: JavaScript variables, control structures, functions, arrays and objects. Cascading Style Sheets, Client Side Scripting - Java Script, PHP: Form processing and business logic, stream processing and regular expressions, viewing client/server environment variables, connecting to database and handling of cookies. SQL, Accessing databases with PHP.

Open Source Ethics: Open source vs. closed source Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source.

Case Studies: Mozilla (Firefox), Wikipedia, Joomla, Open Office, GCC.

Laboratory work: Lab work will primarily focus on installing open source software on different machines. Study and manipulate the source code of different open source OS. To create multimedia using HTML, Explore XHTML, design of XML documents

Recommended Books:

1. Ware B., B Lee J., Open Source Development with Lamp: Using Linux, Apache, MySQL, Perl, and PHP; Addison-Wesley Professional.
2. Rosebrock E.,FilsonE.,Setting Up LAMP: Getting Linux, Apache, MySQL, and PHP Working Together, SYBEX Inc.
3. Deitel, "Internet and World wide web, How to program" 4th Edition, Prentice Hall

COURSE OUTCOMES (COs)

CO1	Learn open source software history, initiatives and principles. Open standards, Licenses and FOSS.
CO2	Learn about the Open Source Operating system and its distributions like Fedora, Google chrome OS, Ubuntu.
CO3	Able to develop web pages, web sites and web applications
CO4	Study of Web technologies based on open Software's LAMP (Linux Apache MySql and PHP/Python)
CO5	Implement HTML, XHTML, PHP and JavaScript

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA202 GRAPH THEORY AND APPLICATIONS

L	T	P	Cr
3	1	0	3.5

Course objective: This course is designed to explore computing and to show students the art of computer programming. Students will learn some of the design principles for writing good programs.

Introduction: Graphs, Sub-graphs, Regular graph, Adjacency and incidence matrices, Finite and infinite graph, Incidence and degree, Isolated vertex, Pendent vertex and null graph, Turan's theorem.

Paths and Circuits: Isomorphism, Walk, Cycle, Paths and circuits, Simple and proper circuit, Connected and disconnected graph, Euler graphs, Operations on graphs, Hamiltonian paths and circuits, Bipartite graph, Berge theorem, Hall's theorem, Edge connectivity, Blocks, Menger's theorem.

Trees and Fundamental Circuits: Trees, Properties of tree, Pendant vertices in a tree, Distance and centers in a tree, Spanning tree, Cayley's Formula, Minimal spanning tree, Prim and Kruskal's algorithm, Matrix Tree theorem, Dijkstra's Shortest Path Algorithm, Floyd-Warshall algorithm, Huffman's Coding Algorithm, Depth-first and breath first algorithm.

Cuts sets and cut-vertices: Cut sets, Properties of cutset, all cut sets in a graph, 1-isomorphism, 2-isomorphism.

Planar graph and dual graphs: Planar graphs, Homoeomorphic graph, Kuratowski's Two Graphs, Different representation of a planar graph, Tutte's f-factor theorem, Detection of planarity, Geometric dual, Combinatorial dual.

Coloring, Covering and Partitioning: Chromatic number, Chromatic Partitioning, Chromatic Polynomial, Covering, Four colour conjecture, Five-colour theorem, Dirac Theorem, Brooks theorem, Vizing theorem.

Directed Graphs: Directed graph, Diagraph and binary relations, Directed Paths, Euler diagraphs, Acyclic digraphs, Topological sorting, Warshall's algorithm, Bellman-Ford algorithm, Ramsey theorems.

Application of Graphs: Study of Konigsberg bridge problem, Travelling-salesman problem, Utilities problem, Electrical network problem, Seating problem, Use of graph in sequential switching networks, Graphs in coding theory, Graphs in computer programming, Flow graph notation, Test case generation using graphs, Job sequencing problem, Graph coloring in scheduling of examinations.

Recommended Books:

1. DeoNarsingh, Graph Theory with applications to Engineering and Computer Science, Prentice-Hall of India (2007).
2. Parmenter Michael M., Goodaire Edgar G., Discrete Mathematics with Graph Theory, Prentice-Hall of India (2007).
3. Rosen, Kenneth H. Discrete Mathematics and its Applications, Tata Mcgraw-Hill (2003).
4. Bondy, J.A. Murty U.S.R., Graph theory and Applications, North Holland Publications (1995).

COURSE OUTCOMES (COs)

CO1	Understand different types of graphs and their relevance to computer science problems
CO2	Able to solve different problems related to trees, circuits and paths
CO3	Able to analyse problems related to partitioning, coloring and covering
CO4	Understand various issues related to directed, undirected Graphs
CO5	Discuss and correlate various Graph applications with computer science and applications

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA 203 OBJECT ORIENTED PROGRAMMING

L	T	P	Cr
3	1	2	4.5

Course objective: Understand fundamentals of object oriented programming in java. To help students understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.

Introduction to Java: History and Evolution of Java,Java Vs. other popular languages, The java programming environment, Fundamental Programming structures: Data types-Primitive Types, Floating-Point Types, Literals, Variables, Type Conversion and Casting, Arrays, Operators-Arithmetic, Bit Wise, Relational, Boolean; Expressions, Statements and Blocks, Control flow statements- Selection, Iteration and Jump Statements.

Object Oriented Programming Concepts in Java: Objects and Classes, declaring Objects, Constructors, This keyword, Method Overloading, Constructor Overloading, Nested classes.

Inheritance: Defining, Applying and Implementing Interfaces; Method Overriding, Super and Final keywords, Polymorphism, Generics, Defining, finding and importing Packages, Exceptions Handling with try, catch, throw, throws and finally keywords, Wrapper classes.

I/O and Threads: Binary I/O, File Handling, Communicating with internet, Thread Model, Creating a thread, Synchronization, Inter-thread Communication, Thread Lifecycle.

Building GUI in Java: Layout Managers, SWING features and Components, Painting, using images, performing animations. Borders, Icons, Event Handling, implementation of Listeners for event handling.

Laboratory work: Main focus is on implementing basic concepts of object oriented programming and to enhance programming skills to solve specific problems.

Recommended books:

1. Deitel H. andDeitel P. , JAVA - How to Program, Pearson Education,(2003).
2. Internet & World Wide Programming, Deitel, Deitel& Nieto, Pearson Education,(2000).
3. Naughton P., Schildt H., JAVA2 – The Complete Reference, Tata McGraw Hill, (2002),5th ed.
4. Greenlaw R. and Hepp G.- Fundamentals of the Internet and the World Wide Web, TMH,(2001).

COURSE OUTCOMES (COs)

CO1	Understand the basics of Internet, E-mail and allied Services.
CO2	Learn the implementation of decision statement and looping statements.
CO3	Understand concepts of Object Oriented Computing in Java.
CO4	Understand Input and output handling from console, files and internet in Java.
CO5	Learn creation of frames, windows, containers, GUI components in Java and event handling for building GUI.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA204 PROGRAMMING LANGUAGES

L	T	P	Cr
3	0	2	4.0

Course Objective: Understanding the concepts of object oriented programming, functional programming, event-driven and reactive programming, formal semantics and language pragmatics.

Object-Oriented Programming: Object-oriented design, Definition of classes: fields, methods, and constructors, Subclasses, inheritance, and method overriding, Dynamic dispatch: definition of method-call, Object-oriented idioms for encapsulation, Using collection classes, iterators, and other common library components

Functional Programming: Processing structured data (e.g., trees) via functions with cases for each data variant, First-class functions (taking, returning, and storing functions), defining higher-order operations on aggregates, especially map, reduce/fold, and filter

Event-Driven and Reactive Programming: Events and event handlers, Canonical uses such as GUIs, mobile devices, robots, servers, Externally-generated events and program generated events, Separation of model, view, and controller

Formal Semantics: Syntax vs. Semantics, Lambda Calculus and Approaches to semantics: Operational, Denotational, Axiomatic, Proofs by induction over language semantics Formal definitions and proofs for type systems, Parametricity, Using formal semantics for systems modelling.

Language Pragmatics: Principles of language design, Evaluation order, precedence, and associativity, Eager vs. delayed evaluation, Defining control and iteration constructs, External calls and system libraries

Laboratory work: Implementing object oriented concepts, Functional Programming concepts, Event Driven Concepts, Reactive Programming concepts and formal semantic concepts.

Recommended Books:

1. Sethi, Ravi, Programming Languages-Concepts and Constructs, Addison-Wesley, 1996, 2nd ed.
2. Tucker A.B. and Noonan R., Programming Languages-Principles and Paradigms, McGraw Hill, 2007, 2nd ed.
3. Sebesta R.W., Concepts of Programming Languages, Addison Wesley, 2012, 10th ed.

COURSE OUTCOMES (COs)

CO1	Compare and contrast different language paradigms and understand object-oriented encapsulation mechanisms such as interfaces and private members.
CO2	Understand the basic algorithms that avoid assigning to mutable state or considering reference equality.
CO3	Understand the event handlers for use in reactive systems, such as GUIs and describe an interactive system in terms of a model, a view, and a controller.
CO4	Use the formal semantics to build a formal model of a software system other than a programming language.
CO5	Iimplement the program whose result can differ under different rules for evaluation order.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA301 COMPUTER NETWORKS

L	T	P	Cr
3	0	2	4.0

Course Objective: For fundamental knowledge of computer networks and related concepts and theories.

Introduction: Organization of the Internet (Internet Service Providers, Content Providers, etc.), Network criteria, Categories of networks, Network performance and Transmission Impairments. Physical pieces of a network, including hosts, routers, switches, bridges, hubs, ISPs, wireless, LAN, access point, and firewalls. OSI Model and TCP/IP Protocol Suite. Layering principles (encapsulation, multiplexing), Switching techniques (e.g., circuit, packet).

Networked Applications : Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.), Distributed applications (client/server, peer-to-peer, cloud, etc.), HTTP as an application layer protocol, Electronic mail, File transfer, Remote login-introduction to protocol specification.

Reliable Data Delivery: Error control (retransmission techniques, timers), Flow control (Acknowledgements, sliding window), Performance issues (pipelining), Process-to-Process Delivery: UDP, TCP and SCTP. Multiplexing with TCP and UDP.

Routing and Forwarding: Routing versus forwarding, Static and dynamic routing, Unicast and Multicast Routing. Distance-Vector, Link-State, Shortest path computation, Dijkstra's algorithm, Network Layer Protocols (IP, ICMP), IP addressing, Address binding with ARP, Scalability issues (hierarchical addressing)

Local Area Networks: LAN topologies: Bus topology, Ring topology, Token passing rings, FDDI, Star topologies, Asynchronous transfer mode. Ethernet, IEEE standards 802.3, 802.5. Multiple Access Problem, Common approaches to multiple accesses (exponential-back off, TDM, etc). Virtual circuit switching including frame relay, X.25, and ATM.

Resource Allocation: Need for resource allocation, Fixed allocation (TDM, FDM, WDM) versus dynamic allocation, End-to-end versus network assisted approaches , Principles of congestion control, Approaches to Congestion, Quality of service, Flow characteristics, Techniques to improve QoS.

Mobility: Principles of cellular networks, Wireless LANs: IEEE 802.11 and Bluetooth, Issues in supporting mobile nodes (home agents).

Recommended Books

1. Forouzan, B.A., Data communication and Networking,McGraw Hill ,(2006) , 4th ed.
2. Tanenbaum ,A.S.,ComputerNetworks,Prentice Hall,(2010) , 5th ed.
3. Kurose and Ross, Computer Networking: A Top Down Approach, Addison-Wesley, (2012) , 6th ed.

4. Stallings, W.,Computer Networking with Internet Protocols and Tech,Prentice Hall of India (2010), 9th ed.

COURSE OUTCOMES (COs)

CO1	Describe and understand the appropriate network applications, terminology and network architecture.
CO2	Discuss services provided by application layer. To define the principles behind naming schemes and resource location.
CO3	Describe the operation of routing protocols, reliable delivery protocols and list the factors affecting the performance of reliable delivery protocols.
CO4	Illustrate forwarding of packets in an IP network and learn various network protocols and algorithms. To list the scalability benefits of hierarchical addressing.
CO5	Describe the Ethernet network and virtual circuit switching. To understand resource allocation, congestion control and mobility.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA302 OPERATING SYSTEMS

L	T	P	Cr
3	0	2	4.0

Course Objective: Role and purpose of the operating system, Functionality of a typical operating system, managing atomic access to OS objects.

Operating System Principles: Structuring methods (monolithic, layered, modular, microkernel models), processes, and resources, Concepts of APIs, Device organization, interrupts: methods and implementations, Concept of user/system state and protection, transition to kernel mode.

Concurrency: Implementing synchronization primitives, Multiprocessor issues (spin locks, reentrancy).

Scheduling and Dispatch: Dispatching and context switching, Preemptive and non-preemptive scheduling, Schedulers and policies, Processes and threads

Memory Management: Review of physical memory and memory management hardware, Working sets and thrashing, Caching, Paging and virtual memory, Virtual file systems.
File Systems: Files: data, metadata, operations, organization, buffering, sequential, nonsequential, Directories: contents and structure, Naming, searching, access, backups, Journaling and log-structured file systems.

Deadlock: Introduction, Analysis of conditions, Prevention & avoidance, Detection & recovery.

Security and Protection: Overview of system security, Security methods and devices, Protection, access control, and authentication.

Virtual Machines: Types of virtualization (including Hardware/Software, OS, Server, Service, Network).

Device Management: Characteristics of serial and parallel devices, Buffering strategies, Direct memory access, Disk structure, Disk scheduling algorithms.

Laboratory Work: To explore different operating systems like Linux, Windows etc. To implement main algorithms related to key concepts in the operating systems using a high level language.

Recommended Books

1. Silberschatz, A., Galvin, P.B. and Gagne, G., Operating System Concepts, John Wiley (2013), 9 th ed.
2. Stallings, Willam, Operating Systems Internals and Design Principles, Prentice Hall, (2014), 7th ed.

3. Dhamdhere, D.M., Operating Systems: A Concept Based Approach, McGraw Hill , (2008), 2nd ed.

COURSE OUTCOMES (COs)

CO1	Able to comprehend the concepts related to basics of operating System.
CO2	Able to exemplify protection and security in operating system.
CO3	Skilled to know and simulate the various process management algorithms and specifications for simple as well as for concurrent processes.
CO4	Able to grasp the concepts related to memory management.
CO5	Able to twig the concepts related to file organization and handling in system.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA303 INFORMATION MANAGEMENT SYSTEM

L	T	P	Cr
3	0	4	5.0

Course objective: Emphasis is on the need of information systems. Main focus is on E-R diagrams, relational database, concepts of normalization and de-normalization and SQL commands.

Introduction: Data, data processing requirement, desirable characteristics of an ideal data processing system, traditional file based system, its drawback, concept of data dependency, Definition of database, database management system, 3-schema architecture, database terminology, benefits of DBMS, Database development process - conceptual data modeling, logical database design, physical database design, database implementation, database maintenance.

Database Analysis: Conceptual data modeling using E-R data model -entities, attributes, relationships, generalization, specialization, specifying constraints. 5 – 6 practical problems based on E-R data model.

Relational Database: Relational data model: Introduction to relational database theory: definition of relation, relational model integrity rules, relational algebra and relational calculus.

Relational Database Design: Normalization- 1NF, 2NF, 3NF, BCNF, 4NF and 5NF. Concept of De-normalization and practical problems based on these forms.

Indexing of Data: Impact of indices on query performance, basic structure of an index, creating indexes with SQL, Types of Indexing and its data structures.

Database Implementation: Introduction to SQL, DDL aspect of SQL, DML aspect of SQL – update, insert, delete & various form of SELECT- simple, using special operators, aggregate functions, group by clause, sub query, joins, co-related sub query, union clause, exist operator.

Laboratory work: Students will learn SQL and other databaseconcepts.One project, which should include database designing & implementation.

Recommended Books:

1. H. F. Korth&Silverschatz, A., Database System Concepts, Tata McGraw Hill, 2010, 6thed.
2. Elmasri&Navathe, Fundamentals of Database Systems, Addison-Wesley, 2011, 6thed.
3. Hoffer, Prescott, Mcfadden, Modern Database Management, Paperback International, 2012, 11th ed.
4. Martin Gruber, Understanding SQL, BPB Publication, 1994, Revised ed.

COURSE OUTCOMES (COs)

CO1	Understand Information Systems as socio-technical systems, Its need and advantages as compared to traditional file based systems.
CO2	Enable students to understand the architecture of DBMS, conceptual data modeling, logical database design and physical database design.
CO3	Enable students to perform Database Analysis using E-R data model by identifying entities, attributes, relationships, generalization and specialization.
CO4	Equip the students to perform Relational Database Design process with Normalization and De-normalization of data and understand the need of Indexing of Data and types of indexes and its implementation.
CO5	Learn Database Implementation with SQL, Usage of DDL aspect of SQL, DML aspect of SQL, aggregate functions, group by clause, sub query, joins, co-related sub query.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA304 DATA STRUCTURES AND ALGORITHMS

L	T	P	Cr
3	0	4	5.0

Course Objectives: To become familiar with different types of Data structures and their applications. To become familiar with different types of algorithmic techniques and strategies.

Linear Data Structures: Arrays, Records, Strings and string processing, References and aliasing, Linked lists, Strategies for choosing the appropriate data structure, Abstract data types and their implementation: Stacks, Queues, Priority queues, Sets, Maps.

Basic Analysis: Differences among best, expected, and worst case behaviours of an algorithm, Asymptotic analysis of upper and expected complexity bounds, Big O notation: formal definition and use , Little o, big omega and big theta notation , Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential, Time and space trade-offs in algorithms, Recurrence relations , Analysis of iterative and recursive algorithms.

Searching and Sorting: Linear Search, Binary Search, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Quick Sort, Heap Sort, Merge Sort, Counting Sort, Radix Sort
.

Algorithmic Strategies with examples and problem solving: Brute-force algorithms with examples, Greedy algorithms with examples, Divide-and-conquer algorithms with examples, Recursive backtracking, Dynamic Programming with examples, Branch-and-bound with examples, Heuristics, Reduction: transform-and-conquer with examples
.

Non-Linear Data Structures And Sorting Algorithms: Hash tables, including strategies for avoiding and resolving collisions , Binary search trees , Common operations on binary search trees such as select min, max, insert, delete, iterate over tree , Graphs and graph algorithms , Representations of graphs, Depth- and breadth-first traversals , Heaps ,Graphs and graph algorithms , Shortest-path algorithms (Dijkstra and Floyd) , Minimum spanning tree (Prim and Kruskal).

Problem Clauses: P, NP, NP- Hard and NP-complete, deterministic and non-deterministic polynomial time algorithm approximation and algorithm for some NP complete problems. Introduction to parallel algorithms, Genetic algorithms, intelligent algorithms
.

Laboratory Work: Implementation of Arrays, Recursion, Stacks, Queues, Lists, Binary trees, Sorting techniques, Searching techniques. Implementation of all the algorithmic techniques.

Recommended Books:

1. Corman, Leiserson&Rivest, Introduction to Algorithms, MIT Press(2009), 3rdedition.
2. Langsam, Y. and Augenstein, M.J., Data Structures Using C and C++,Dorling Kindersley, 2008, 2nd ed.

3. Trembley, J.P., Sorenson, P.G., An introduction to data structures with applications, Tata McGraw Hill, 2008, 2nd ed.
4. Sahni, Sartaj, Data Structures, Algorithms and Applications in C++, Universities Press 2005, 2nd ed.

COURSE OUTCOMES (COs)

CO1	Implement and learn the basic data structures and learn the appropriate algorithmic approach to a problem and implement various search and sorting techniques.
CO2	Learn various algorithmic strategies.
CO3	Learn the appropriate algorithmic approach to a problem and implement various search and sorting techniques. Solve problems using fundamental algorithms.
CO4	Demonstrate the ability to evaluate algorithms, to provide justification for that selection, and to implement the algorithm in a particular context
CO5	Learn deterministic and non-deterministic polynomial time algorithm approximation and algorithm for some NP complete problems.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA305 SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

L	T	P	Cr
3	0	2	4.0

Course Objectives: To apply principles of software development and evolution. To specify, abstract, verify, validate, plan, develop and manage large software and learn emerging trends in software engineering.

Introduction: Introduction to Software Engineering, importance of Software, The Software Evolution, Software Characteristics, Software Applications, Software Crisis: Problem and Causes.

Software Processes: Software process models (Waterfall, Incremental, and Evolutionary process models and Agile), software quality concepts, process improvement, software process capability maturity models, Personal Software process and Team Software Process, Overview of Agile Process.

Tools and Environments: CASE and its Scope, CASE support in Software Life Cycle, Documentation Support, Architecture of CASE Environment.

Requirements Engineering: Problem Analysis, Requirement elicitation and Validation, Requirements modeling: Scenarios, Information and analysis classes, flow and behavioral modeling, documenting Software Requirement Specification (SRS).

Software Design: System design principles: levels of abstraction (architectural and detailed design), separation of concerns, information hiding, coupling and cohesion, Structured design (top-down functional decomposition), object-oriented design, event driven design, component-level design, data-structured centered, aspect oriented design , function oriented, service oriented, Design patterns.

Software Construction: Coding Practices: Techniques, mechanisms for building quality programs including Secure Coding Practices, Integration Strategies, Internal Documentation, Verification.

Software Verification and Validation: Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Specification, Software Testing Strategies, Verification & Validation, Unit, Integration Testing, Top Down and Bottom Up Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Debugging.

Software Quality Assurance: Software Quality Control and Quality Assurance, ISO 9000 Certification for Software Industry, SEI CMM and Comparison of ISO & SEICMM.
Software Evolution: Software development in the context of large, pre-existing code bases, Software evolution, Characteristics of maintainable software, Software Reengineering, Software reuse.

Formal Methods: Role of formal specification and analysis techniques in the software development cycle. Formal approaches to Software Modeling and Analysis.

Technical Metrics for Software: A Framework for Technical Software Metrics, Metrics for the Analysis Model, Metrics for Design Model, Metrics for Source Code, Testing and Maintenance.

Laboratory Work: Implementation of Software Engineering concepts using tools like Rational Suite etc. Exposure to CASE tools like Rational Software suit, Turbo Analyst, Silk Suite.

Recommended Books:

1. S. Pressman R., Software Engineering, McGraw Hill International Ed, 2014 8th ed.
2. Sommerville I., Software Engineering, Addison-Wesley Publishing, (2010) 9th ed.
3. Jalote P., An integrated Approach to Software Engineering, Narosa (2005).
4. James F. Peter, Software Engineering - An Engineering Approach, John Wiley (2004).

COURSE OUTCOMES (COs)

CO1	Able to comprehend the concepts related to theoretical foundation of Software and its Engineering.
CO2	Able to identify a suitable process model among given set of process models.
CO3	Able to gather, identify and specify the underlying complete details of a system and its related domains using different approaches and formal methods.
CO4	Articulate design principles and event based designing.
CO5	Understand software verification & validation approaches and Software Quality Assurance.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA401 AUTOMATA THEORY AND COMPUTABILITY

L	T	P	Cr
3	1	0	3.5

Course Objective: This course introduces basic theory of computer science and formal methods of computation. The course exposes students to the computability theory, as well as to the complexity theory.

Regular Languages: Alphabets, Language, Regular Expression, Definitions of Finite State Machine, Transition Graphs, Deterministic & Non-deterministic Finite State Machines, Regular Grammar, Thompson's Construction to Convert Regular Expression to NDFA & Subset Algorithm to convert NDFA to DFA, Various recent development in the Conversion of Regular Expression to NFA, Minimization of DFA, Finite State Machine with output-Moore machine and Melay Machine, Conversion of Moore machine to Melay Machine & Vice-Versa.

Properties of Regular languages: Conversion of DFA to Regular Expression, Pumping Lemma, Properties and Limitations of Finite state machine, Decision properties of Regular Languages, Application of Finite Automata.

Context Free Grammar and Push Down Automata : Context Free Grammar,Derivation tree and Ambiguity, Application of Context free Grammars, Chomsky and Greibach Normal form, Properties of context free grammar, CKY Algorithm, Decidable properties of Context free Grammar, Pumping Lemma for Context free grammar, Push down Stack Machine, Design of Deterministic and Non-deterministic Push-down stack.

Turing Machine: Turing machine definition and design of Turing Machine, Church-Turing Thesis, Variations of Turing Machines, combining Turing machine, Universal Turing Machine, Post Machine, Chomsky Hierarchy, Post correspondence problem.

Uncomputability: Halting Problem, Turing enumerability, Turing Acceptability and Turing decidabilities, unsolvable problems about Turing machines, Rice's theorem.

Recommended Books:

1. Hopcroft J. E., Ullman J. D. and Motwani R., Introduction to Automata Theory, Languages and Computation, Pearson Education, 3rd Edition, (2006).
2. John C. Martin, Introduction to Languages and the Theory of Computation, McGraw-Hill Higher Education (2011).
3. Daniel A. Cohen, Introduction to Computer Theory, John Wiley and Sons, 2nd Ed, 1996
4. Michael Sipser, Introduction to the Theory of Computation, Thomson, 2nd Ed, 2007

COURSE OUTCOMES (COs)

CO1	Explore regular languages and finite automata
CO2	Explore context-free languages, push-down automata.
CO3	Get a broad overview of the theoretical foundations of computer science.
CO4	Familiarize with thinking analytically and intuitively for problem-solving situations in related areas of theory in computer science
CO5	Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs and mastering concepts of Turing Machine.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA402 GRAPHICS AND VISUAL COMPUTING

L	T	P	Cr
3	0	2	4.0

Course Objective: Detailed study of computer graphics, 2 D and 3 D transformations, representations and visualization.

Fundamentals of Computer Graphics: Applications of computer Graphics in various, Video Display Devices, Random scan displays, raster scan displays, DVST, Flat Panel displays, I/O Devices.

Graphics Primitives: Algorithms for drawing Line, circle, ellipse, arcs & sectors, Boundary Fill & Flood Fill algorithm, Color Tables

Transformations: 2D & 3D Scaling, Translation, rotation, shearing & reflection, Composite transformation, Window to View port transformation.

Clipping: Cohen Sutherland, Liang Barsky, Nicholl - Lee - Nicholl Line clipping algorithms, Sutherland Hodgeman, Weiler Atherton Polygon clipping algorithm.

Three Dimensional Object Representations: 3D Modeling transformations, Parallel & Perspective projection, Clipping in 3D. Curved lines & Surfaces, Spline representations, Spline specifications, Bezier Curves & surfaces, B-spline curves & surfaces, Rational splines, Displaying Spline curves & surfaces.

Basic Rendering: Rendering in nature, Polygonal representation, Affine and coordinate system transformations, Visibility and occlusion, depth buffering, Painter's algorithm, ray tracing, forward and backward rendering equations.

Visualization: Visualization of 2D/3D scalar fields: color mapping, isosurfaces. Direct volume data rendering: ray-casting, transfer functions, segmentation. Visualization of: Vector fields and flow data, Time-varying data, High-dimensional data: dimension reduction, parallel coordinates, Non-spatial data: multi-variate, tree/graph structured, text Perceptual and cognitive foundations, Evaluation of visualization methods, Applications of visualization.

Laboratory work: Lab work should be done in OpenGL. Covers all the basic drawing, filling, transformation and clipping algorithms.

Recommended Books:

1. Donald D Hearn, M. Pauline Baker, Computer Graphics C version, Pearson Education, 2nd ed.
2. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis, 5th Edition, 2013
3. James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes, Computer Graphics: Principles & Practice in C, Addison Wesley Longman, 2nd ed.

4. Zhigang Xiang, Roy A Plastock, Computer Graphics, Schaums Outline, TMH, 2nd ed.

COURSE OUTCOMES (COs)

CO1	Able to comprehend the concepts related to basics of computer graphics and visualization.
CO2	Skilled to know and simulate the various graphics transformations and clipping techniques.
CO3	Able to grasp the concepts related three dimensional object representations.
CO4	Learn and program in OpenGL.
CO5	Able to understand the methods, applications of 2D and 3D visualization.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA403 MOBILE APPLICATION DEVELOPMENT

L	T	P	Cr
3	0	2	4.0

Course objective: This course is concerned with the development of applications on mobile and wireless computing platforms.

Introduction: Cost of Mobile Application Development, Importance of Mobile Strategies, Challenges, Myths, Third-Party Frameworks, Mobile Web Presence, Applications

Introduction to Mobility: Mobility Landscape, Mobile Platforms**,** Mobile apps development, Overview of Android Platform, Setting up the mobile apps development environment with emulator.

Building block of Mobile apps: App user Interface Designing, Layout, User Interface elements, Draw-able, Menu, Activity states and lifecycle, Interaction among activities.
App functionality based user interface:Threads, Asynchronous task, Services-states and lifecycle, Notifications, Broadcast receivers, Telephony and SMS API**.**

Naïve Data Handling: On Device File I/O, Shared preferences, Mobile Databases such as SQLite and enterprise data access.

Sprucing up Mobile Apps: Graphics and animation-custom views, canvas, animation API multimedia-audio/video playback and record, location aware.

Testing Mobile apps: Debugging Apps, White and Black Box Testing and test automation of apps.

Creating Consumable Web Services for Mobile Devices: What is a Web Service, Web Services Languages (Formats), Creating an Example Web Service, Debugging Web Services

Mobile User Interface Design: Effective Use of Screen Real Estate, Understanding Mobile Information Design, Understanding Mobile Application Users, Understanding Mobile Platforms, Using the Tools of Mobile Interface Design.

Mobile Websites: Choosing a Mobile Web Option, Adaptive Mobile Websites, Dedicated Mobile Websites Mobile Web Apps with HTML5

Android: Android as Competition to itself, Connecting to the Google Play, Android Development Practices, Building an App in Android

iOS: IOS Project, Debugging iOS Apps, Objective-C Basics, Building the Derby App in IOS

Windows Phone 7: Windows Phone 7 Project, Building an App in Windows Phone 7, Distribution

Laboratory work: To develop robust mobile applications and work on related tools and technologies. exploring the application development for different mobile platforms like Android, iPhone, Symbian

Recommended Books:

1. Professional Mobile Application Development, Jeff Mcwherter, Scott Gowell, Wrox Publisher, 1st Ed. 2012
2. Sams Teach Yourself Android Application Development in 24 Hrs, Lauren Darcy and Shane Conder, 1sted.
3. Mobile Application Security, HimanshuDwivedi, Chris Clark, David Thiel, Tata McGraw Hill, 1st Edition, 2010

COURSE OUTCOMES (COs)

CO1	Understand the concept of mobility landscape, mobile apps development and mobile app development environment along with emulator
CO2	Design an app user interface with the use of provided GUI resources
CO3	Learn about the life cycle, states and interaction among the activities.
CO4	Understand the working of Threads, Services, Notifications and Broadcast Receivers, on device file IO and Shared preferences.
CO5	Design the animated and multimedia oriented mobile apps by the use of animation API.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA404 BIG DATA ANALYTICS AND BUSINESS INTELLIGENCE

L	T	P	Cr
3	1	2	4.5

Course Objective: To have a basic understanding of most recent advancements in Big Data and using insights, statistical models, visualization techniques for its effective application in Business intelligence.

Introduction to Data Analytics: Data and Relations, Data Visualization, Correlation, Regression, Forecasting, Classification, Clustering.

Big Data Technology Landscape: Fundamentals of Big Data Types, Big data Technology Components, Big Data Architecture, Big Data Warehouses, Functional vs. Procedural Programming Models for Big Data.

Introduction to Business Intelligence: Business View of IT Applications, Digital Data, OLTP vs. OLAP, Why, What and How BI? , BI Framework and components, BI Project Life Cycle, Business Intelligence vs. Business Analytics.

Big Data Analytics: Big Data Analytics, Framework for Big Data Analysis, Approaches for Analysis of Big Data, ETL in Big Data, Introduction to Hadoop Ecosystem, HDFS, Map-Reduce Programming, Understanding Text Analytics and Big Data, Predictive analysis on Big Data, Role of Data analyst.

Business implementation of Big Data: Big Data Implementation, Big Data workflow, Operational Databases, Graph Databases in a Big Data Environment, Real-Time Data Streams and Complex Event Processing, Applying Big Data in a business scenario, Security and Governance for Big Data, Big Data on Cloud, Best practices in Big Data implementation, Latest trends in Big Data, Latest trends in Big Data, Big Data Computation, More on Big Data Storage, Big Data Computational Limitations.

Laboratory Work: Introduction to most recent advancements in Big Data technology along with their usage and implementation with relevant tools and technologies.

Recommended books:

1. Michael Minelli, Michele Chambers, AmbigaDhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley CIO Series(2013),1st ed.
2. T. white, Hadoop: The Definitive Guide, O' Reilly Media (2012), 3rd ed.

COURSE OUTCOMES (COs)

CO1	Translate a business challenge into an analytics challenge and deploy a structured lifecycle approach to data science and big data analytics projects.
CO2	Usage of Big Data in present World.
CO3	Analyze big data, create statistical models, and identify insights that can lead to actionable results.
CO4	Visualization techniques, communicate analytic insights to business sponsors, and others and explain how advanced analytics can be leveraged to create competitive advantage.
CO5	Define and distinguish a data scientist from a traditional business intelligence analyst.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA 405 ARTIFICIAL INTELLIGENCE

L	T	P	Cr
3	0	2	4.0

Course Objective: To be familiar with the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, and learning methods in solving particular engineering problems.

Fundamental Issues: Overview of AI problems, Examples of successful recent AI applications, Intelligent behaviour, The Turing test, Rational versus non-rational reasoning, Problem characteristics: Fully versus partially observable, Single versus multi-agent, Deterministic versus stochastic, Static versus dynamic, Discrete versus continuous, Nature of agents: Autonomous versus semi-autonomous, Reflexive, Goal-based, and Utility-based, Importance of perception and environmental interactions, Philosophical and ethical issues.

Basic Search Strategies: Problem spaces (states, goals and operators), Problem solving by search, Factored representation (factoring state into variables), Uninformed search (breadth-first, depth-first, depth-first with iterative deepening), Heuristics and informed search (hill-climbing, generic best-first, A*), Space and time efficiency of search, Constraint satisfaction (backtracking and local search methods).

Advanced Search: Constructing search trees, Dynamic search space, Combinatorial explosion of search space, Stochastic search: Simulated annealing, Genetic algorithms, Monte-Carlo tree search, Implementation of A* search, Beam search, Minimax Search, Alpha-beta pruning, Expectimax search (MDP-solving) and chance nodes.

Knowledge Representation: Propositional and predicate logic, Resolution in predicate logic, Question answering, Theorem proving, Semantic networks, Frames and scripts, conceptual graphs, conceptual dependencies.

Reasoning under Uncertainty: Review of basic probability, Random variables and probability distributions: Axioms of probability, Probabilistic inference, Bayes' Rule, Conditional Independence, Knowledge representations using Bayesian Networks, Exact inference and its complexity, Randomized sampling (Monte Carlo) methods (e.g. Gibbs sampling), Markov Networks, Relational probability models, Hidden Markov Models, Decision Theory Preferences and utility functions, Maximizing expected utility.

Agents: Definitions of agents, Agent architectures (e.g., reactive, layered, cognitive), Agent theory, Rationality, Game Theory Decision-theoretic agents, Markov decision processes (MDP), Software agents, Personal assistants, and Information access Collaborative agents, Information-gathering agents, Believable agents (synthetic characters, modelling emotions in agents), Learning agents, Multi-agent systems Collaborating agents, Agent teams, Competitive agents (e.g., auctions, voting), Swarm systems and Biologically inspired models.

Expert Systems: Architecture of an expert system, existing expert systems: MYCIN, RI.

Expert system shells.

Laboratory work: Programming in C/C++/java: programs for Search algorithms- Depth first, breadth first, best first, hill climbing, Implementation of games: 8-puzzle, Tic-tac-toe using heuristic search, Designing expert system using logic in prolog, implementing an intelligent agent.

Recommended Books

1. Rich E., Artificial Intelligence, Tata McGraw Hills (2009) 3rd ed.
2. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education Asia (2009) 6th ed.
3. Patterson D.W, Introduction to AI and Expert Systems, Mc GrawHill 1998, 1st ed.
4. ShivaniGoel, Express Learning- Artificial Intelligence, Pearson Edu Asia (2013), 1st ed.

COURSE OUTCOMES (COs)

CO1	Learn and understand applications of artificial intelligence and categorize various problem domains and search methods
CO2	Explore and understand advanced search techniques and algorithms like minimax for game playing
CO3	Learn knowledge representation using various forms in artificial intelligence and understand the importance of probability in knowledge representation for reasoning under uncertainty.
CO4	Learn to represent knowledge using Bayesian networks and drawing Hidden Markov Models.
CO5	Learn the architecture for intelligent agents and implementing an intelligent agent and an expert system.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA602: ADVANCED APPLICATION DEVELOPMENT

L	T	P	Cr
2	0	2	3.0

Course Objectives: Students will able to understand the .NET framework, concepts of C# programming and ASP.NET technology. Students can enable to develop Integrated Development environment to build GUI based, Data Base based applications in .NET Framework using C#..

Basic C# programming: Introduction to the .NET Framework 4.0- CLR, BCL, CLS, CTS; Features of C#, Datatypes, Operators, Flow control statements, Enumerations, Arrays: n-dimensional, Jagged arrays, Methods, Param array, String and stringbuilder, Exception handling.

OOP's based C# programming: OOP's concepts, Classes, Namespaces, Constructors and destructors, Garbage collection, Boxing and un-boxing, Operator overloading, Properties, Indexers, Inheritance, Abstract classes and interfaces, Structures, Collections and Generics, Class library(DLL) and Assemblies, Delegates, Multithreading, File handling- Stream reader, Stream writer, Binary reader, Binary writer, Text reader, Text writer, File methods; Serialization and formatters.

GUI and ADO.Net: Events, Windows forms: Common controls, Container, Menu and toolbars, Data and dialog controls, Connection object, Command object, Datareader, Dataset, Dataadapter, Connected and disconnected scenarios, GUI components for data control and data binding, Stored procedure, XML data access, Language integrated query(LINQ).

ASP.net: Introduction, Architecture, working with web form and server controls, Client side and server side validation, enhancing web sites using master pages and themes, State management, Data connectivity using ASP.net, Silverlight application.

Laboratory Work: The lab work will be based on the C# programming concepts, console applications, window forms, database connectivity using ADO.net and ASP.net, XML, web services and server controls .net technologies.

Recommended Books

1. Dietel H.M., Deitel P.J., Hoey P.R. and Yaeger C.H., Simply C#: an application-driven tutorial approach, Pearson Education (2003).
2. Schildt H., C# 4.0 The Complete Reference, Tata Mc-Graw Hill (2010).
3. Hejlsberg A., Torgersen M., Wiltamuth S., and Golde P., The C# Programming Language, Microsoft (2010).
4. Robinson S., Nagel C., Watson K.,GlynnJ.,Skinner M. and Evjen B., Professional C# 4.0 and .NET 4, Wrox (2010).
5. Hanselman S., Rader D. and Evje B. ASP.NET 4 in C# and VB, Wrox (2010).

COURSE OUTCOMES (COs)

CO1	Able to Compare and contrast the .Net Framework with Java technologies
CO2	Able to Design window based, web based applications.
CO3	Able to Develop applications using latest technologies of .Net Framework.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA603: ADVANCE JAVA PROGRAMMING

L	T	P	Cr
2	0	4	4.0

Course Objectives: Objective of this course is to provide the ability to design console based, GUI based and web based applications. Students will also be able to understand integrated development environment to create, debug and run multi-tier and enterprise-level applications

Introduction: Features, Java basics- identifiers, variables, data types, operators, control structures, arrays, jagged arrays, command line arguments; Object oriented programming- classes, objects, inheritance, method overriding and hiding, interface, abstract class, polymorphism, inner class, wrapper classes, Boxing, Packages; Exception handling, Collections, Generics, File I/O, Serialization, Multi threading.

GUI and JDBC: Event handling , AWT Controls, Window forms and controls, Layout managers, Menus, Applet, JFC, JDBC- Drivers and architecture, Connection object, Types of Statement, Stored procedures.

Servlets: Introduction, Servlet life cycle, Deployment and web.xml, Servlet chaining, Session management, Cookies, Web context, Init and context parameter, Authentication and concurrent access, Servlet listeners, Creating war files.

JSP and JSTL: Architecture, Life cycle, JSP tags, Expressions, JSP with database, Implicit objects, Expression language, Exception handling in JSP, Session management, Directives, Using java beans, Tag libraries, Miscellaneous tags, Custom tags- bodyless, body tags.

EJB and Struts: Types of enterprise beans, Life cycle, Session beans, Entity beans, Message driven beans, JNDI, Hibernate, Struts architecture, Struts classes, Action mapping, Struts flow, Combining Struts and tiles, Introduction to spring framework.

Lab Work: Lab assignments will be based on object oriented features of Java, Window forms, JDBC, Servlets, JSP, Java Beans, EJB, Hibernate, struts and springs.

Recommended Books

1. Herbert Schildt, Java - The Complete Reference, Tata McGraw- Hill, Seventh Edition(2008).
2. Jim Keogh, J2EE- The Complete Reference; Tata Mcgraw-Hill, Edition(2002).
3. Alur Deepak, Malks Dan and Crupi John, **Core J2EE Patterns: Best Practices and Design Strategies,** Prentice Hall India (2001).
4. Austin and Pawlan, Advanced Programming for JAVA 2 Platform, Pearson Education (2004).
5. Geary M. David , Core JSTL Mastering the JSP standard Tag Library, Pearson Education(2007).

COURSE OUTCOMES (COs)

CO1	Able to Use graphical interface to create more efficient Java programs.
CO2	Able to Implement web applications using Java servlets and JSP
CO3	Able to Create web based applications EJB, Hibernate, Struts and Spring framework.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA604 : ENTERPRISE RESOURCE PLANNING

L	T	P	Cr
2	0	2	3.0

Course Objectives: To make student able to build an understanding of the fundamental concepts of ERP systems, their architecture, and working of different modules in ERP. Students will also able to develop and design the modules used in ERP systems, and can customize the existing modules of ERP systems.

Introduction to ERP: ERP Overview, Benefits, Business process reengineering, ERP implementation life cycle, Options of various paradigms, Supply chain Management, Critical factors guiding selection and evaluation, Strategies for successful implementation, impediments and initiatives to achieve success, Critical success and failure factors, Integrating ERP into organizational culture.

SAP and ABAP: Architecture of SAP, Data types in ABAP, ABAP programming Language, ABAP User Dialogs, Function groups and function modules, Accessing Database Access, open SQL, Native SQL, ABAP Object Orientation, Classes and objects in ABAP, Inheritance, Interfaces, Triggering and Handling Events, ABAP data dictionary, Declarations, selection screens, Formatting and Displaying Data, Program Events, , Dynpros, BSP applications.

SD: Basic functions and master data in SD, Sales orders, Deliveries, Pricing, Billing, Transportation, Credit Management.

MM: Basic functions and master data, Consumption based planning, Purchasing, Inventory management, Evaluation of materials, Invoice verification, Balance sheet evaluation,Material ledger.

Laboratory Work: Implementation of ERP concepts using any ERP package.

Recommended Books

1. Mary Sumner, Enterprise Resource Planning, Pearson Education (2010).
2. Kogent Learning Solutions Inc., SAP ABAP/4 Covers SAP ECC 6.0 Black Book, Dreamtech Press (2009).
3. Bradford M., Modern ERP Systems: Select Implement and Use Today's Advanced Business Systems (2010).
4. Dreamtech Software Team, SAP Architecture, Administration, Basis, ABAP Programming with MM and SD modules, Dreamtech Press (2005).

COURSE OUTCOMES (COs)

CO1	Able to Understand the architecture of ERP and working of different modules in ERP.
CO2	Able to Identify the risks and challenges in implementation of ERP system
CO3	Able to Develop and design the ERP modules and also to Customize the existing modules of ERP systems.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA 501: NETWORK SECURITY

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course is designed to impart a critical theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools.

Introduction: Security, Functionality and ease of use Triangle, Essential Terminology, Elements of Security, Difference between Penetration Testing and Ethical Hacking, Deliverables ethics and legality, Computer Crimes and Implications.

Reconnaissance: Information Gathering Methodology, Locate the Network Range, Active and Passive reconnaissance

Scanning: Scanning, Elaboration phase, active scanning, scanning tools NMAP, hping2. Enumeration, DNS Zone transfer. Detecting live systems on the target network, Discovering services running /listening on target systems, Understanding port scanning techniques, Identifying TCP and UDP services running on the target network, Understanding active and passive fingerprinting

Trojans and Backdoors: Effect on Business, Trojan, Overt and Covert Channels, Working of Trojans, Different Types of Trojans, Different ways a Trojan can get into a system, Indications of a Trojan Attack, Some famous Trojans and ports used by them

Sniffers: Definition of sniffing, Sniffer working, Passive Sniffing, Active Sniffing, Ethreal tool, Man-in-the-Middle Attacks, Spoofing and Sniffing Attacks, ARP Poisoning and countermeasures.

Denial of Service: Goal of DoS (Denial of Service), Impact and Modes of Attack.

Social Engineering: Social Engineering, Art of Manipulation, Human Weakness, Common Types of Social Engineering, Human Based Impersonation, Example of Social Engineering, Computer Based Social Engineering, Reverse Social Engineering, Policies and Procedures, Security Policies-checklist

Session Hijacking: Understanding Session Hijacking, Spoofing vs Hijacking, Steps in Session Hijacking, Types of Session Hijacking, TCP Concepts 3 Way and shake, Sequence numbers

Ethical Hacking- System Hacking and Hacking Wireless Networks: Aspect of remote password guessing, Role of eavesdropping ,Various methods of password cracking, Keystroke Loggers, Understanding Sniffers ,Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing. Introduction to 802.11, Role of WEP, Cracking WEP Keys, Sniffing Traffic, Wireless DOS attacks, WLAN Scanners, WLAN Sniffers, Hacking Tools, Securing Wireless Networks.

Laboratory work: To learn and experimentdifferent types of scenerios related to Security of an organization. To implement core algorithms in high level language and to use related tools and technologies for various security concepts above.

Recommended Books

1. Hackers Beware, Eric Core, EC-Council Press, 2003
2. Network Security Essentials, William Stallings ,Prentice Hall, 5th Edition, 2013
3. Firewalls and Internet Security, William R. Cheswick and Steven M. Bellovin, Addison-Wesley Professional, 2nd Edition, 2003.
4. Cryptography and Network Security, W. Stallings , Prentice Hall, 5th Edition, 2010

COURSE OUTCOMES (COs)

CO1	Demonstrate knowledge of various vulnerabilities in network applications.
CO2	Awareness of various malicious content and guiding ways for protection against the same.
CO3	Demonstrate knowledge of various forms of attacks.
CO4	Familiarization about judicious use of various tools.
CO5	Expertise in the techniques of ethical practices to test the system vulnerabilities.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA502 PARALLEL AND DISTRIBUTED COMPUTING

L	T	P	Cr
3	0	2	4

Course Objective: To learn the concepts of Parallel and Distributed Computing and its implementation for assessment of understanding the course by the students

Parallelism Fundamentals : Scope and issues of parallel and distributed computing, Parallelism, Goals of parallelism, Parallelism and concurrency, Multiple simultaneous computations, Programming Constructs for creating Parallelism, communication, and coordination. Programming errors not found in sequential programming like data races, higher level races, lack of liveness

Parallel Architecture : Architecture of Parallel Computer, Communication Costs, parallel computer structure, architectural classification schemes, Multicore processors, Memory Issues : Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, vector processing, GPU, co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA)

Parallel Decomposition and Parallel Performance: Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task-Based Decomposition, Data Parallel Decomposition, Actors and Reactive Processes, Load balancing, Data Management, Impact of composing multiple concurrent components, Power usage and management. Sources of Overhead in Parallel Programs, Performance metrics for parallel algorithm implementations, Performance measurement, The Effect of Granularity on Performance Power Use and Management, Cost-Performance trade-off
;
Communication and Coordination: Shared Memory, Consistency, Atomicity, Message-Passing, Consensus, Conditional Actions, Critical Paths, Scalability,cache coherence in multiprocessor systems, synchronization mechanism.

CUDA programming model: Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on the parallel computing device, API function to transfer data to parallel computing device, Concepts of Threads, Blocks, Grids, Developing kernel function that will be executed by threads in the parallelized part, Launching the execution of kernel function by parallel threads, transferring data back to host processor with API function call.

Parallel Algorithms design, Analysis, and Programming : Parallel Algorithms, Parallel Graph Algorithms, Parallel Matrix Computations, Critical paths, work and span and relation to Amdahl's law, Speed-up and scalability, Naturally parallel algorithms, Parallel algorithmic patterns like divide and conquer, map and reduce, Specific algorithms like parallel Merge Sort, Parallel graph algorithms, parallel shortest path, parallel spanning tree, Producer-consumer and pipelined algorithms

Laboratory Work: To implementparallel programming using CUDA with emphasis on developing applications for processors with many computation cores, mapping computations to parallel hardware, efficient data structures, paradigms for efficient parallel algorithms

Recommended Books

1. C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley 2008.
2. A Grama, AGupra, G Karypis, V Kumar. Introduction to Parallel Computing (2nd ed.). Addison Wesley, 2003.
3. B Gaster, L Howes, D Kaeli, P Mistry, and D Schaa. Heterogeneous Computing With Opencl. Morgan Kaufmann and Elsevier, 2011.
4. T Mattson, B Sanders, B MassingillPatterns for Parallel Programming. Addison-Wesley 2004.
5. Quinn, M. J. (2004). Parallel Programming in C with MPI and OpenMP. McGraw-Hill.

COURSE OUTCOMES (COs)

CO1	Familiarize the concepts of issues and tasks in parallel and distributed computing, different parallel architectures, programming models, and algorithms for common operations.
CO2	Develop an efficient parallel algorithm to solve it.
CO3	Analyze algorithms time complexity as a function of the problem size and number of processors
CO4	Aanalyze its code performance, determine computational bottlenecks, and optimize the performance of the code
CO5	Use CUDA programming to solve different parallel algorithms

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA503 HUMAN COMPUTER INTERACTION

L	T	P	Cr
3	0	2	4.0

Course Objective: To understand the interaction between human and computer components and design interactive user interface.

Introduction to graphical user interface: Importance of user Interface – definition, importance of good design, benefits of good design, a brief history of Screen design, Popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

Design process: Interaction design basics, Human interaction with computers, design rules, implementation support, Evaluation techniques, Importance of human characteristics, Human consideration, Human interaction speeds, and understanding business junctions
.

Models and Theories: Cognitive Models- Goal and task hierarchies, Linguistic Models, Task analysis- task decomposition, knowledge based analysis, Entity-relationship-based techniques, Uses of task analysis.

Screen Designing: Design goals, Screen planning and purpose, organizing screen elements, ordering of screen data and content , screen navigation and flow , Visually pleasing composition , amount of information, focus and emphasis , presenting information simply and meaningfully , information retrieval on web , statistical graphics , Technological consideration in interface design.

Menus, Navigation and Windows: Structure of menus, Functions of menus, Navigating menus, Types of graphical menus, Navigation schemes for selection of window, window characteristics, components of a window, window presentation styles, types of windows, window management, organizing window functions, window operations.

Components and Interaction Devices: Text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.Keyboard and function keys, pointing devices, speech recognition digitization and generation, image and video displays, drivers
.

User-Centered Design and Testing: Functionality and usability requirements, Techniques for gathering requirements, Techniques and tools for the analysis and presentation of requirements, Prototyping techniques and tools, Evaluation without users, using both qualitative and quantitative techniques.

Laboratory Work: Main focus is on designing and implementing visually appealing graphical user interface. To implement all the related programs on relevant tools and technologies.

Recommended Books:

1. Wilbert O. Galitz, The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, John Wiley & Sons (2007).
2. Shneidermann B., Designing the user interface, Pearson Education Asia (2004), 3rd ed.
3. Dix A., Fincay J., Goryd G., Abowd, Bealg R., Human Computer Interaction, Pearson Education (2004), 3rd ed.
4. Lauesen S., User Interface Design: A Software Engineering Perspective, Pearson Education (2004).

COURSE OUTCOMES (COs)

CO1	Understand the interaction between human and computer components, importance of good interface design.
CO2	Implement the principles of GUI and design a visually pleasing UI.
CO3	Understand and implement interaction design basics and associated design rules, cognitive architecture, linguistic models and task analysis techniques including task decomposition.
CO4	Understand the structure of menus, navigational scheme and characteristics of windows, concepts of Multimedia, pointing devices, speech recognition digitization and generation.
CO5	Learn gathering requirements and usage of tools for analysis and user requirements.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA504 WEB ANALYTICS AND INTELLIGENCE

L	T	P	Cr
3	0	2	4.0

Course Objective: This subject enables to examine how website visitors view and interact with a site's pages and features, and business intelligence, which would allow using data on customer purchasing patterns, demographics, and demanding trends to make effective strategic decisions.

Introduction: Definition, Process, Key terms: Site references, Keywords and Key phrases; building block terms: Visit characterization terms, Content characterization terms, Conversion metrics; Categories: Offsite web, On site web; Web analytics platform, Web analytics evolution, Need for web analytics, Advantages, Limitations.

Data Collection: Clickstream Data: Web logs, Web Beacons, JavaScript tags, Packet Sniffing; Outcomes Data: E-commerce, Lead generation, Brand/Advocacy and Support; Research data: Mindset, Organizational structure, Timing; Competitive Data: Panel-Based measurement, ISP-based measurement, Search Engine data.

Qualitative Analysis: Heuristic evaluations: Conducting a heuristic evaluation, Benefits of heuristic evaluations; Site Visits: Conducting a site visit, Benefits of site visits; Surveys: Website surveys, Post-visit surveys, Creating and running a survey, Benefits of surveys.

Web Analytic fundamentals: Capturing data: Web logs or JavaScripts tags, Separate data serving and data capture, Type and size of data, Innovation, Integration, Selecting optimal web analytic tool, Understanding clickstream data quality, Identifying unique page definition, Using cookies, Link coding issues.

Web Metrics: Common metrics**:** Hits, Page views, Visits, Unique visitors, Unique page views, Bounce, Bounce rate, Page/visit, Average time on site, New visits; Optimization(e-commerce, non e-commerce sites): Improving bounce rates, Optimizing adwords campaigns; Real time report, Audience report, Traffic source report, Custom campaigns, Content report, Google analytics, Introduction to KPI, characteristics, Need for KPI, Perspective of KPI, Uses of KPI.

Web analytics 2.0: Web analytics 1.0, Limitations of web analytics 1.0, Introduction to analytic 2.0, Competitive intelligence analysis : CI data sources, Toolbar data, Panel data ,ISP data, Search engine data, Hybrid data, Website traffic analysis: Comparing long term traffic trends, Analyzing competitive site overlap and opportunities.

Google Analytics: Brief introduction and working, Adwords, Benchmarking, Categories of traffic: Organic traffic, Paid traffic; Google website optimizer, Implementation technology, Limitations, Performance concerns, Privacy issues.

Relevant technologies: Internet & TCP/IP, Client / Server Computing, HTTP (HyperText Transfer Protocol), Server Log Files & Cookies, Web Bugs.

Laboratory Work: Lab assignments cover the basic fundamentals of analyzing the websites, understanding their basic features. It also involves use of related tools and technologies in this domain.

Text books:

1. Clifton B., Advanced Web Metrics with Google Analytics, Wiley Publishing, Inc. (2010), 2nd ed.
2. Kaushik A., Web Analytics 2.0 The Art of Online Accountability and Science of Customer Centricity, Wiley Publishing, Inc. (2010),1st ed.

3. Sterne J., Web Metrics:Proven methods for measuring web site success, John Wiley and Sons (2002),1st ed

COURSE OUTCOMES (COs)

CO1	Understand the basic concepts of web analytics, its goals, key terminologies, advantages, limitations, various data collection methods for the purpose of analysis.
CO2	Performing qualitative analysis using heuristic evaluation, site visits and surveys. To explore and understand fundamentals of web analytics.
CO3	Explore various web metrics and understand what metrics to optimize, understanding key performance indicators, their role, and how to implement them for success.
CO4	Understand the limitations of web 1.0. To explore web analytic 2.0.
CO5	Understand what is Google analytics and how it works.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA505 DATABASE ADMINISTRATION

L	T	P	Cr
3	0	2	4.0

Course Objective: Detailed study of database in all aspects from database administration, management, creation, security, configuring the DBMS network to data backup, recovery and its transportation.

Database administration: Database, Data, and System Administration. DBA tasks, Types of DBA, New Technology Impacts on DBA. DBA roles and responsibility.

Database Architectural Components: Database architectures, DBMS Server, DBMS Instance, System Global Area, Program Global Area, Process Structure, Background Processes, Logical Structure, Installing and managing DBMS servers.

Creating the Database and Its Management: Planning the Database, Managing the database Instance, Starting and Stopping Database Control, Database Startup and Shutdown database Change Management: Managing Database Storage Structures- How Table Data Is Stored, Creating a New Table space, Altering a Table space, Actions with Table spaces, Dropping Table spaces, Managing Schema Objects-Managing data files, Managing tables, indexes, and constraints, Managing Data and Concurrency, Managing Undo Data.

Data Availability, Administering User and Database Security: Define availability, Availability Problems, Impact of Change on Database Structures, Performance Management: System, Database, and Application performance. Database User Accounts, Predefined Accounts: SYS and SYSTEM, Creating a User, Authenticating Users, Privileges, Benefits of Roles, Creating a Role, Profiles and Users, Assigning Quota to Users, Implementing Database Security.

Configuring the DBMS Network Environment: Database net server side configuration, basic net client-side configuration, usage and configuration of Database shared server architecture. Handling Metadata.

Data Backup, Recovery and Its transportation: Backup and Recovery, Disaster Planning, Data and storage management. Transporting data between databases - export and import utility, Loading data into database-SQL*loader. DBA Tools

Laboratory work: To understand and use database environment, physical structure, granting/revoking privileges and RMAN. To use tools and techniques related to data base administration.

Recommended Books:

1. Oracle Database 10*g*:Administration Workshop I, Oracle Press
2. Loney K., Baryla B. Oracle database 10g: DBA handbook, Tata McgrawHills.
3. Thomas B., Baryla B., Oracle DBA Fundamental-II, BPB publications.

4. Carig S. Mullins, Database Administration: The Complete Guide to DBA Practices and Procedures Second Edition, Addison-Wesley.

COURSE OUTCOMES (COs)

CO1	Understand Database Server Architectural Components.
CO2	Learn to Create a Database and Its Management.
CO3	Understand of administration of User and Database Security.
CO4	Configure Database Network Environment.
CO5	Perform Data Backup, Recovery and Its transportation.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
4.	MST	20
5.	EST	40
6.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA 506 CLOUD INFRASTRUCTURE AND SERVICES

L	T	P	Cr
3	0	2	4.0

Course Objective: To learn the concepts of cloud infrastructure and services and its implementation for assessment of understanding the course by the students.

Introduction and Evolution of Computing Paradigms: Overview of Existing Hosting Platforms, Cluster Computing, Grid Computing, Utility Computing, Autonomic Computing, mesh, Introduction to Cloud Computing, Cloud Computing history and evolution, practical applications of cloud computing for various industries, economics and benefits of cloud computing.

Cloud Issues and Challenges: Cloud computing issues and challenges like Security, Elasticity, Resource management and scheduling, QoS (Quality of Service) and Resource Allocation, Cost Management, Big Data.

Data Center : Classic Data Center, Virtulaized Data Center (Compute, Storage, Networking and Application) , Business Continuity in VDC

Cloud Computing Architecture: Cloud Architecture model, Types of Clouds: Public Private & Hybrid Clouds, Cloud based services: Iaas, PaaS and SaaS

Classification of Cloud Implementations: Amazon Web Services, The Elastic Compute Cloud (EC2), The Simple Storage Service (S3), The Simple Queuing Services (SQS), Google AppEngine - PaaS, Windows Azure, Aneka, A Comparison of Cloud Computing Platforms .

Virtualization: Virtualization, Advantages and disadvantages of Virtualization, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, VMware vCloud – IaaS

Cloud based Data Storage: Introduction to Map Reduce for Simplified data processing on Large clusters, Design of data applications based on Map Reduce in Apache Hadoop, Task Partitioning, Data partitioning, Data Synchronization, Distributed File system, Data Replication , Shared access to weakly consistent to data stores, introduction to Python.

Laboratory Work: To implement Cloud, Apache and Hadoop framework and related services. To understand various concepts practically about virtualization, data storage. To implement few algorithms with the help of MapReduce and some high level language.

Recommended Books:

1. Raj Kumar Buyya, James Broberg, AndrezeiM.Goscinski, Cloud Computing: Principles and paradigms, 2011
2. Michael Miller, Cloud Computing, Que Publishing 2008
3. Cloud Computing: A practical Approach Anthony Velte, Toby Velte and Robert Elsenpeter by Tata McGrawHill

4. Judith Hurwitz, Robin Bllor, Marcia Kaufman, F Halper, Cloud Computing for dummies, 2009.

COURSE OUTCOMES (COs)

CO1	Learn the concept of Existing Hosting Platforms and computing paradigms currently being used in industry and academia
CO2	Learn the principles, practices and advantages of Cloud Computing and its evolution and features.
CO3	Familiarize the issues related to Cloud Computing. To analyse IASS/ PAAS and SAAS services along with Cloud models.
CO4	Learn the concepts of various Cloud Platforms with comparative analysis and the concepts of virtualization with the advantages in Cloud.
CO5	Understand the Map reduce framework of Hadoop and design python programs for deploying database applications in cloud computing.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA507 USER CENTERED DESIGN

L	T	P	Cr
3	0	2	4.0

Course Objective: Learn the methods and concepts of analyzing user-centred requirements which are the basis for designing software interfaces and Web sites that make sense to the user.

Introduction to UCA*:* Mental models, Knowing how the user works, Which UCA steps to perform, Obstacles of user-centred analysis, ROI justifying the right process.

Creating a Design Strategy: Components of a design strategy, Site strategy drives design strategy, Where to get strategy information, mining existing documentation, working with brand objectives.

Profiles and Personas*:* The value of profiles and personas, User profiles, Task profiles, Environment profiles, Personas. Data gathering methods, choosing the right method, Elements of field interview, developing interview questions, Good and bad interview technique, conducting a user observation.

Complementary Data Gathering Methods*:* Value of complementary methods, Focus groups, User group meetings and usability roundtables, Facilitated workshops and JAD sessions, Using surveys and other indirect methods, Online surveys, Using multiple methods.

Scenario and Task Analysis*:* The power of a scenario, Scenarios vs. use cases, Determining the level of detail, Scenarios drive priorities, Identifying functions and tasks, Common errors and challenges in task analysis, Characterizing the new task design.

Primary Noun Architecture*:* Value of primary nouns, Identifying primary nouns, Describing primary nouns, Primary noun views, Defining primary noun details, primary nouns to navigation.

Information Architecture*:* Costs of poor organization, Basic organization schemes, Hybrid schemes, Shallow vs. deep structures, Labeling systems, Affinity diagrams and card sorting techniques, Card sorting tools, getting sign-off on the contract for design, Using concept sketches to drive out requirements, Setting usability criteria.

Laboratory Work: To evaluate and fix task flows, develop design strategies, user profiles, high level task. Creation of task prioritization, environmental profile and high level navigation diagrams.

Text Books:

1. Buxton, “Sketching User Experiences”, San Francisco: Morgan Kaufmann, 2007.
2. Moggridge.B, Designing Interactions, 1 edition, Cambridge, MA: The MIT Press, 2007.
3. Travis Lowdermilk, “user centred design: A Developer’s Guide”, 2013.
4. Courage, C. and K. Baxter, “Understanding Your Users”, 1 ed,Morgan Kaufmann; 2005.
5. Righi, C and J. James , “User centred design stories”, 2007.

COURSE OUTCOMES (COs)

CO1	Learn the knowing how the user works.
CO2	Understand various Components of a design strategy.
CO3	Understand the value of profiles and personas.
CO4	Aanalyses and explore various data gathering methods and to understand the Value of complementary methods.
CO5	Determine the level of detail, and explore the Value of primary nouns.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA508 IMAGE PROCESSING AND VIDEO PROCESSING

L	T	P	Cr
3	0	2	4.0

Course Objective: To learn the advanced concepts Image Processing and Video Processing and its implementation for assessment of understanding the course by the students.

Introduction and Digital Image Fundamentals: Digital Image Representation, Fundamental steps in Image Processing, Elements of Digital image processing systems, Sampling and quantization, neighbors of a pixel, adjacency, connectivity, Regions and Boundaries, Distance measures, Image operations on a pixels basis, Linear and Nonlinear operations

Image Enhancement and Restoration: Image Enhancement in the Spatial domain: Gray level transforms, Histogram Processing, Enhancement using Arithmetic/Logic Operations, smoothing and sharpening filters, Image Enhancement in the Frequency domain: 1-D and 2-D Fourier Transform and their Inverse, Filtering, Smoothing and sharpening domain filters, Homomorphic Filtering, Degradation Model, Noise models, Restoration in the presence of Noise only spatial filtering, Periodic Noise reduction by frequency domain filtering, estimating the degradation function

Color Image Processing: Color models, Pseudocolor Image Processing, Color Transforms, Smoothing and sharpening, Color Segmentation, Noise in color images, Color Image compression

Image Compression: Fundamentals, Compression Models, Error free Comparison, Lossy Compression, wavelets in Image compression, Image compression standards. Morphological Image Processing: Dilation and Erosion, Basic Morphological algorithms, Extension to gray scale images.

Image Segmentation, Representation and Description: Detection of discontinuities, Edge linking and boundary detection, Thresholding, Region oriented Segmentation, Motion based Segmentation; Representation schemes, Boundary description, Regional descriptors, Morphology.

Object Recognition: Patterns and Pattern classes, Decision Theoretic Methods, Structural methods.

Video Processing: Video formation, Video sampling, Video modeling, 2-D & 3-D Motion estimation, Motion-compensated (MC) filtering, Deinterlacing, Frame-rate conversion, Video Watermarking, Video Coding, Video Compression, Frame-based compression (MPEG-1/2), Scalable or layered frame-based compression, Object-based compression (MPEG-4), Error control in video communications.

Laboratory Work: The lab work will be based on image enhancement, image zooming, image cropping, image restoration, image compression and image segmentation etc.

Recommended Books:

1. Gonzalez R. C. and Woods R. E., "Digital Image Processing", Edition 2nd, Pearson Education, 2007.
2. Jain A. K., "Fundamentals of Digital Image Processing", Edition 4th, PHI, 2002.
3. Umbaugh S E, "Computer Imaging, Digital Image Analysis and Processing", Edition 3rd, CRC Press Book, 2005.
4. Gonzalez R C, Woods R.E. and Eddins S.L., "Digital Image Processing using MATLAB", Pearson Education, 2004.
5. Wang Y, Ostermann J, Zhang Y Q, "Video Processing and Communications", Edition 1st, Prentice Hall, 2001.

COURSE OUTCOMES (COs)

CO1	Learn steps of image processing using popular image processing techniques.
CO2	Understand concepts of Image Restoration techniques like noise and degradation models.
CO3	Learn many Image compression techniques for both gray and colored images.
CO4	Attain knowledge of Image segmentation techniques.
CO5	Enable comprehension of object recognition, representation of images and videos.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
4.	MST	20
5.	EST	40
6.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA509 COMPILER CONSTRUCTION

L	T	P	Cr
3	0	2	4.0

Course Objective: Gain the working knowledge of the major phases of compilation. Develop the ability to use formal attributed grammars for specifying the syntax and semantics of programming languages. Learn about function and complexities of modern compilers and design a significant portion of a compiler.

Introduction to compiling: Compilers, Analysis of the source program, the phases of Compiler, Compilation and Interpretation, Bootstrapping and Cross compiler.

Lexical Analysis: Need of Lexical analyzer, Tokens and regular expressions, Generation of lexical analyzer from DFA, Introduction to LEX and program writing in LEX.

Syntax Analysis: Need for syntax analysis and its scope, Context free grammar, Top down parsing, bottom up parsing, backtracking and their automatic generation, LL(1) Parser, LR Parser, LR(0) items, SLR(1), LALR(1), Canonical Parsing, Introduction to YACC and Integration with LEX.

Error Analysis: Introduction to error analysis, detection, reporting and recovery from compilation errors, Classification of error-lexical, syntactic and semantic with examples, Detection of syntactic error in LL and LR parsers, panic mode error recovery and error recovery in YACC tool.

Static semantics and Intermediate Code generation: Need for various static semantic analyses in declaration processing, name and scope analysis, S-attribute def. and their evaluation in different parsing, Semantic analysis through S-attribute grammar, L-attribute def. and their evaluation.

Run time Environment: Need for runtime memory management, Address resolution of runtime objects at compile time, Type checking, Language features influencing run time memory management, Parameter passing mechanism, Division of memory into code, stack, heap and static, Activation record, Dynamic memory management, garbage collection.

Code Generation: Code generation for expressions, Issues in efficient code generation, Sethi Ullman algorithm, Dynamic programming approach for optimal code generation tree, Introduction to retarget able code generation, code generation for control structures.

Code Optimization: Need for code optimizations, Local and global optimization, Control flow analysis, Data flow analysis, performing global optimizations, Graph colouring in optimization, Live ranges of run time values.

Laboratory work: To implementing concepts of various phases of a compiler in a high level language. To implement a mini working compiler.

Text Books:

1. Aho A.V., Ullman J. D., Sethi R., Compilers Principles, Techniques and Tools, Pearson Education, Edition 2nd ,2005
2. John Levine, Tony Mason, Doug Brown, Lex and Yacc, O'Reilly, Edition 2nd1992
3. Kenneth C. Louden, Compiler Construction and Practices, Thomson Publication, Edition 2nd, 1997.
4. Dhamdhere, Compiler Construction. Macmillan Publication Edition 2nd , 2008

COURSE OUTCOMES (COs)

CO1	Attain knowledge of the major phases of compilation, particularly lexical analysis, parsing, semantic analysis, and code generation.
CO2	Ability to use of formal attributed grammars for specifying the syntax and semantics of programming languages.
CO3	Ability to design and implement a significant portion of a compiler for a language chosen by the instructor.
CO4	Develop an awareness of the function and complexity of modern compilers
CO5	Exposure to aspects of theoretical computer science including Languages, Grammars, and Machines.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

MCA 510 AGILE SOFTWARE DEVELOPMENT APPROACHES

L	T	P	Cr
3	0	2	4.0

Course Objectives: To learn the fundamental principles and practices associated with each of the agile development methods. To apply the principles and practices of agile software development on a project of interest and relevance to the student.

Agile Software Development: Basics and Fundamentals of Agile Process Methods, Values of Agile, Principles of Agile, stakeholders, Challenges

Lean Approach: Waste Management, Kaizen and Kanban, add process and products add value. roles related to the lifecycle, differences between Agile and traditional plans, differences between Agile plans at different lifecycle phases. Testing plan links between testing, roles and key techniques, principles, understand as a means of assessing the initial status of a project/ How Agile helps to build quality

Agile and Scrum Principles: Agile Manifesto, Twelve Practices of XP, Scrum Practices, Applying Scrum. Need of scrum, working of scrum, advanced Scrum Applications, Scrum and the Organization , scrum values

Agile Product Management: Communication, Planning, Estimation Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement, Escalating issue. Quality, Risk, Metrics and Measurements, Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement and Escalating issue

Agile Requirements: User Stories, Backlog Management. Agile Architecture**:** Feature-Driven Development. Agile Risk Management**:** Risk and Quality Assurance, Agile Tools

Agile Testing: Agile Testing Techniques, Test-Driven Development, User Acceptance Test

Agile Review: Agile Metrics and Measurements, The Agile approach to estimating and project variables, Agile Measurement, Agile Control: the 7 control parameters. Agile approach to Risk, The Agile approach to Configuration Management, The Atern Principles , Atern Philosophy ,The rationale for using Atern, Refactoring, Continuous integeration, Automated Build Tools

Scaling Agile for large projects: Scrum of Scrums, Team collaborations, Scrum, Estimate a Scrum Project, Track Scrum Projects, Communication in Scrum Projects, Best Practices to Manage Scrum.

Laboratory Work: exploring the tools related to Agile Development and approached and develop small projects using this technology.

Recommended Books

1. Agile Software Development, Principles, Patterns, and Practices (Alan Apt Series) Robert C. Martin (Author) ,2011
2. Succeeding with Agile : Software Development Using Scrum, Pearson 2010

COURSE OUTCOMES (COs)

CO1	Analyze existing problems with the team, development process and wider organization
CO2	Apply a thorough understanding of Agile principles and specific practices
CO3	Select the most appropriate way to improve results for a specific circumstance or need
CO4	Judge and craft appropriate adaptations to existing practices or processes depending upon analysis of typical problems
CO5	Evaluate likely successes and formulate plans to manage likely risks or problems

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	40
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40