Education for a World Stage

NM INSTITUTE OF ENGINEERING & TECHNOLOGY

Sijua, Patrapada, Near AIIMS , Bhubaneswar-751019,Odisha

SYLLABUS

2 Yrs Master in Computer Application (MCA)

from the Admission Batch

2021-22

BIJU PATNAIK UNIVERSITY OF TECHNOLOGY, ODISHA

ROURKELA

Curriculum and Syllabus

2 Yrs Master in Computer Application (MCA)

from the Admission Batch

2021-22

First Semester							
Theory							
Sl. No.	Category	Course Code	Course Title	L-T-P	Credit	University Marks	Internal Evaluation
1	BS	MCA01001	Discrete Mathematics	3-0-0	3	100	50
2	PC	MCA01002	Computer System Architecture	3-0-0	3	100	50
3	PC	MCA01003	C and Data Structure	3-0-0	3	100	50
4	PC	MCA01004	Operating System	3-0-0	3	100	50
5	PC	MCA01005	Database Engineering	3-0-0	3	100	50
Total Credit (Theory)					**15**		
Total Marks						**500**	**250**
Practical							
1	PC	MCA01006	Data Structure Using C Lab	0-0-3	2		100
2	PC	MCA01007	Operating System Lab	0-0-3	2		100
3	PC	MCA01008	Database Engineering Lab	0-0-3	2		100
Total Credit (Practical)					**6**		
Total Semester Credit					**21**		
Total Marks							**300**

DISCRETE MATHEMATICS

UNIVERSITY COURSE CODE: MCA01001

COURSE OUTCOME:

Students will be able to

CO1	Outline Sets and their algebra, duality, power sets and partitions, Principle of Strong Mathematical Induction and Product sets.
CO2	Explain Solving problems using Recurrence Relations, Pigeon-Hole Principle and Invertible Functions.
CO3	Solve various numeric and generating functions.
CO4	Analyze solution of recurrence relations by the method of generating functions, divide and conquer algorithms
CO5	Evaluate problems using groups, Rings and Boolean Algebra (Lattice, Principle of duality etc.)
CO6	Design graphs and trees to formulate solutions for real life problems.

Module-I **(10 Hours)**

Logic: Propositional equivalence, predicates and quantifiers, Methods of proofs, proof strategy, sequences and summation, mathematical induction, recursive definitions and structural induction, program correctness, prepositional calculus.
Counting: The basics of counting, the pigeonhole principle, permutations and combinations, recurrence relations, solving recurrence relations, generating functions, inclusion-exclusion principle, application of inclusion-exclusion.

Module-II **(10 Hours)**

Relations: Relations and their properties, n-array relations and their applications,representing relations, closure of relations, Warshall's algorithm, equivalence of relations, partial orderings.
Graph theory: Introduction to graphs, graph terminology, representing graphsand graph isomorphism, connectivity, Euler and Hamilton paths, planar graphs, graph coloring, introduction to trees, application of trees.

Module-III **(06 Hours)**

Group theory: Groups, subgroups, generators and evaluation of powers, cosets and Lagrange's theorem, permutation groups and Burnside's theorem, isomorphism, auto orphisms, homomorphism and normal subgroups, rings, integral domains and fields.

Module-IV **(08 Hours)**

Lattice theory: Lattices and algebras systems, principles of duality, basic properties of algebraic systems defined by lattices, distributive and complimented lattices, Boolean lattices and Boolean algebras, uniqueness of finite Boolean expressions.

Module-V **(06 Hours)**

Coding theory: Coding of binary information and error detection, decoding anderror correction.

Content beyond Syllabus:

1. Permutation and Combination.
2. Binomial Theorem.
3. Coding and Decoding.
4. Set theory and its application

Text Book:

1. C. L. Liu, D. P. Mohapatra, Elements of Discrete Mathematics: A computer Oriented Approach, McGraw Hill Education (India) Private Limited, 4th Edition, 2013.

Reference Books:

1. R.K.Bisht, and H.S.Dhami, Discrete Mathematics, Oxford University Press, First Edition, 2015
2. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw Hill, 5thed, 2003.
3. J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications, to Computer Science, TataMc-Graw Hill, 2001.
4. Joe L. Mott, A. Kandel, and T. P. Baker, Discrete Mathematics for Computer Scientists & Mathematics, Prentice Hall of India, 2nd Edition, 2006.
5. N. Deo, Graph Theory with applications to Engineering & Computer Science, Prentice Hall of India, 2006.
6. S. Lipschutz, Discrete Mathematics, Tata McGraw Hill, 2005

COMPUTER ORGANIZATION AND ARCHITECTURE

UNIVERSITY COURSE CODE: MCA01002

COURSE OUTCOME:

Students will be able to

CO1	Understand the functional unit of the processor such as the register file and arithmetic logical unit and with the basic of system topic
CO2	Evaluate cost-performance & design trade-offs in designing and constructing a computer processor.
CO3	Design flowchart for Concurrent access to memory and cache coherency in Parallel Processors and describe the process.
CO4	Evaluate CPU organization and instruction & a memory module and analyze its operation by interfacing with the CPU.
CO5	Apply and Implement fundamental coding schemes & draw the functional block diagram and describe the function of the instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.
CO6	Create the organization for the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit

Module – I: **(08 Hours)**

Introduction: Review of basic computer architecture, Quantitative techniques in computer design, measuring and reporting performance.

Module – II: **(08 Hours)**

Pipelining: Basic concepts, Instruction and Arithmetic pipeline, Data hazards, Control hazards and Structural hazards, Techniques for handling hazards. Exception handling. Pipeline optimization techniques.

Module – III: **(08 Hours)**

Hierarchical memory technology: Inclusion, Coherence and locality properties, Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, Mapping and Management techniques, Memory replacement policies.

Module – IV: **(08 Hours)**

Instruction-level Parallelism: Basic concepts, Techniques for increasing ILP, Superscalar, Super pipelined and VLIW Processor architectures. Array and Vector processors

Module – V: **(08 Hours)**

Multiprocessor architecture: Taxonomy of Parallel Architectures, Centralized shared- memory architecture, Synchronization, Memory consistency, Interconnection networks. Distributed shared memory architecture. Cluster computers

Content beyond Syllabus:

1. Advanced Processor Architectures.
2. Parallel Processing
3. Secure Hardware Design.
4. Pipelining and Superscalar Architectures
5. Instruction Set Architectures (ISAs).

Text Books:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, Fifth Edition, Tata McGraw Hill, 2002.
2. William Stallings, “Computer Organization and Architecture – Designing for Performance”, Sixth Edition, Pearson Education, 2003.
3. Patterson, “Computer Organization and Design”, Elsevier
4. John P Hayes, “Computer Organization”, McGraw Hill 5. Morris Mano,” Computer System Architecture”, PHI

C & DATA STRUCTURE

UNIVERSITY COURSE CODE: MCA01003

COURSE OUTCOME:

Students will be able to

CO1	Understand the basic concepts of the C programming language.
CO2	Develop algorithms to solve simple to complex problems using C.
CO3	Implement data structures (e.g., arrays, linked lists, stacks, queues) in C.
CO4	Apply various searching and sorting techniques on data structures.
CO5	Demonstrate knowledge of dynamic memory allocation and pointers in C.
CO6	Student should be capable of designing, implementing and analyzing algorithms to solve various problems

MODULE – I **(10 Hours)**

C Language Fundamentals, Arrays and Strings

Character set, Identifiers, Keywords, Data Types, Constant and Variables, Statements, Expressions, Operators, Precedence of operators, Input – output Assignments, Control structures, Decision making and Branching, Decision making & looping. Declarations.

MODULE – II **(10 Hours)**

Monolithic vs Modular programs, User defined vs standard functions, formal vs Actual arguments, Functions category, function prototypes, parameter passing, Recursion, Storage Classes: Auto, Extern, Global, Static. Character handling in

C. String handling functions. Pointers, Structures, Union & File handling

MODULE – III **(10 Hours)**

Pointer variable and its importance, Pointer Arithmetic passing parameters, Declaration of structures, pointer to pointer, pointer to structure, pointer to function, unions dynamic memory allocations, unions, file handling in C.

MODULE – IV **(10 Hours)**

Development of Algorithms: Notations and Analysis, Storage structures for arrays-sparse matrices, Stacks and Queues: Applications of Stack: Prefix, Postfix and Infix expressions. Circular queue, Double ended queue.

Content beyond Syllabus:

1. Memory Management: Deep dive into memory management techniques, memory leaks, memory fragmentation in C programming.
2. Greedy Algorithms: Exploring greedy algorithm strategies, applications, and their proofs of optimality.
3. File Streams: In-depth understanding of file streams, binary file I/O, and custom serialization Techniques.

Text Books:

1. E. Balagurusamy, Programming in ANSI 'C', 8th Edition, Tata McGraw Hill, 2019.
2. Reema Thareja, Data Structures Using C, 2nd Edition ,Oxford University Press, 2014.
3. M. Tanenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd.

Reference Books:

1. A.K.Rath and A. K. Jagadev, "Data Structures and Program Design using C", 2nd Edition, Scitech Publications, 2011.
2. Bruno R Preiss, "Data Structures and Algorithms with Object Oriented Design Pattern in C++", John Wiley & Sons, Inc., 1999.
3. Horowitz and Sahani, "Fundamentals of data Structures",Galgotia Publication Pvt. Ltd.

OPERATING SYSTEM

UNIVERSITY COURSE CODE: M C A 0 1 0 0 4

COURSE OUTCOME:

Students will be able to

CO1	Explain the structure and functions of Operating system.
CO2	Illustrate the concept of concurrency.
CO3	Analyze processes, threads and scheduling algorithms.
CO4	Outline the concepts of deadlock.
CO5	Distinguish between various memory management scheme.
CO6	Explain I/O management and file system, concepts of protection and security.

MODULE-I

(08 Hours)

Overview of Operating Systems: Introduction, how OS takes System Control, Why OS is essential, Functions of the Operating Systems, Evolution of Operating Systems, Generations of OS.

MODULE-II **(08 Hours)**

Operating System Structure & Processes: Introduction, System Components, Operating System Structure, Operating System Services, System Calls, System Programs, Process, Process States, Process Control.

MODULE-III **(08 Hours)**

Operating System Services for Process Management & Scheduling: Introduction, Process Creation, Termination & Other Issues, Threads, Multithreading, Types of Threads, Schedulers, Types of Schedulers, Types of Scheduling, Scheduling Algorithms, Types of Scheduling Algorithms.

MODULE-IV **(08 Hours)**

Process Synchronization, Interprocess Communication & Deadlock: Introduction, Data Access and Control Synchronization, Critical Sections, Race Condition, Classical Problems & Solutions of Process Synchronization, Semaphores, Message Passing, Deadlock, Conditions for Deadlock, Resource Allocation Graph, Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlocks.

MODULE-V **(08 Hours)**

Memory Management & Virtual Memory: Introduction, Memory Management Schemes, Sharing and Protection in Paging, Sharing and Protection in Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Thrashing

Content beyond Syllabus:

1. Real time operating system
2. Android OS Architecture.
3. Cryptography technique.
4. Hard real time OS

Text Books:

1. Silberschatz and Galvin, “Operating System Concepts”, John Wiley Publishing
2. William Stallings, “Operating Systems Internals & Design Principles”, Pearson Education
3. Naresh Chauhan, “Principles of Operating Systems”, Oxford India Publications

References:

4. Pabitra Pal Choudhury, “Operating System Principles and Design”, PHI Publication
5. Sibsankar Halder and Alex A. Aravind, “Operating System”, Pearson Education

DATABASE ENGINEERING

UNIVERSITY COURSE CODE: MCA01005

COURSE OUTCOME:

Students will be able to

CO1	Understand the basic concepts of data base systems and identify different data base Architecture schemas, data models.
CO2	Illustrate the components of E R model and describe the storage architecture.
CO3	Demonstrate relational algebra, relational calculus and apply them in database design..
CO4	Analyze and illustrate the different normal forms and classify their utility in data base design.
CO5	Determine the properties of transaction processing, concurrency control and timestamp methods.
CO6	Design and Create relational database from different case studies and formulate the uses of distributed database, parallel database, data ware housing, data mining & Big data.

Module -I **(06 Hours)**

Introduction to DBMS: concept and overview of DBMS, data models, DB languages, DB users and Administrator, 3-schema architecture of DBMS, data independence, EF Codd Rule.

Module -I I **(06 Hours)**

ER Model: basic concepts, design issues, keys, ER diagram, Weak entity sets, Extended ER features. Relational model: structure of relational model, Relational algebra, extended relational algebra Operations.

Module – III **(08 Hours)**

Relational database design: FDs, Anamolies in designing DB, Normalization using FDs, various Normal forms-1NF, 2NF, 3NF, BCNF, 4NF, 5NF.

Module-IV **(10 Hours)**

SQL and Integrity Constraints: Concepts of DDL, DML, DCL, various SQL operations: set operations, aggregate functions, constraints, view, nested sub queries, PL/SQL, cursor, trigger.

Module – V **(10 Hours)**

Internals of RDBMS: Query optimization, various optimization algorithms, Transaction processing, concurrency control and recovery management. Advanced Database: OODB, WEB based DB, Data warehousing and Data mining.

Content beyond Syllabus

1. JDBC Connectivity using database
2. Normalization criteria and optimize queries
3. Design object based databases-ODL,OQL

Text Books:

1) Korth, Silverschatz, Abraham," Database system concepts", Tata McGraw Hill Publication
2) R.Elmasri, S.B Navathe, "Fundamentals of Database System", Adision Wesley Publishing
3) Er.Rajiv chopra, "Database management systems, A Practical Approach", S.Chand Publishing
4) Ramkrishna, "Database management systems", Tata McGraw Hill Publication

DATA STRUCTURE USING C LAB

UNIVERSITY COURSE CODE: MCA01006

COURSE OUTCOME:

Students will be able to

CO1	Explain the basic data structures and their applications and to analyze the time and space complexities of algorithms (knowledge)
CO2	Choose appropriate data structures to represent data items in real world problems
CO3	Design data structures using various trees and arrange them in an optimal way using heap
CO4	Analyze and implement various kinds of searching and hash techniques
CO5	Identify the proper path by using BFS ,DFS and different searching techniques
CO6	Identify the proper path by using BFS ,DFS and different searching techniques

LIST OF EXPERIMENTS:

1. Implementation of Stack Using Array.
2. Implementation of Queue Using Array.
3. Implementation of Infix to Postfix Conversion using Stack.
4. Evaluation of Postfix Expression using Stack.
5. Implementation of Singly Linked List.
6. Implementation of Doubly Linked List.
7. Implementation of Stack Using Linked List.
8. Implementation of Queue Using Linked List.

9. Implementation of Binary Tree Traversal: Preorder, In order and Post order.
10. Implementation of Binary Search Tree.
11. Implementation of sorting algorithms: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort and Heap sort.
12. Implementation of Searching Algorithms : Linear Search and Binary Search
13. Implementation of Breadth First Search (BFS) in a Graph.
14. Implementation of Depth First Search (DFS) in a Graph.
15. Implementation of Hashing using hash functions.

Experiment beyond Syllabus:

1. Design circle, rectangle and some simple graphics using Graphics. Library
2. Create a database in .csv format

OPERATING SYSTEM LAB

UNIVERSITY COURSE CODE: MCA01007

COURSE OUTCOME:

Students will be able to

CO1	Implement various CPU scheduling algorithms
CO2	Implement various page replacement algorithms
CO3	Explain the process of system calls.
CO4	Apply the various file operations.
CO5	Implement various disk scheduling algorithms
CO6	Implement various classical problem

LIST OF EXPERIMENTS:

1. Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.

a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority

2. Write a C program to simulate Multi-level Feedback Queue Scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – System processes and User processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

3. Write a C program to simulate the MVT and MFT memory management techniques.

4. Write a C program to simulate the following Contiguous Memory allocation techniques

a) Worst-fit b) Best-fit c) First-fit

5. Write a C program to simulate Paging technique of Memory management.

6. Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.

7. Write a C program to simulate Disk scheduling algorithms a) FCFS b) SCAN c) C-SCAN

8. Write a C program to simulate Page replacement algorithms a) FIFO b) LRU c) LFU

9. Write a C program to simulate Page replacement algorithms a) Optimal

10. Write a C program to simulate Producer-Consumer problem using semaphores.

11. Write a C program to simulate the concept of Dining-Philosophers problem.

Experiment beyond Syllabus:

1. Paging techniques
2. Guest operating system concept like VMware

DATABASE ENGINEERING LAB

UNIVERSITY COURSE CODE: MCA01008

COURSE OUTCOME:

Students will be able to

CO1	Develop database modeling for a problem.
CO2	Design a database using normalizations
CO3	Implement a database query language.
CO4	Develop GUI using front end tool
CO5	Develop a connection between frontend and database
CO6	Implement a Data Manipulation Language

LIST OF EXPERIMENTS:

1. Execute a single line and group functions for a table.
2. Execute DCL and TCL Commands.
3. Create and manipulate various DB objects for a table.
4. Create views, partitions and locks for a particular DB
5. Write PL/SQL procedure for an application using exception handling
6. Write PL/SQL procedure for an application using cursors.
7. Write a DBMS program to prepare reports for an application using functions.
8. Write a PL/SQL block for transaction operations of a typical application using triggers.

9. Write a PL/SQL block for transaction operations of a typical application using package.
10. Design and develop an application using any front end and back end tool (make use of ER diagram and DFD).
11. Create table for various relation.
12. Implement the query in sql for a) insertion b) retrieval c) updating d) deletion.
13. Creating Views
14. Writing Assertion
15. Writing Triggers
16. Implementing operation on relation using PL/SQL
17. Creating Forms
18. Generating Reports

Experiment beyond Syllabus:

1. Creation of databases triggers
2. Database design of E-R model

Second Semester							
Theory							
Sl. No.	Category	Course Code	Course Title	L-T-P	Credit	University Marks	Internal Evaluation
1	PC	MCA02001	Computer Networks	3-0-0	3	100	50
2	PC	MCA02002	Analysis and Design of Algorithms	3-0-0	3	100	50
3	PC	MCA02003	Object Oriented Programming Using Java	3-0-0	3	100	50
4	PC	MCA02004	Object Oriented Analysis & Design	3-0-0	3	100	50
5	PC	MCA02005	Internet and Web Programming	3-0-0	3	100	50
Total Credit (Theory)					**15**		
Total Marks						**500**	**250**
Practical							
1	PC	MCA02006	Java and Python Programming Lab	0-0-3	2		100
2	PC	MCA02007	Computer Networks Lab	0-0-3	2		100
3	PC	MCA02008	Algorithm Design Lab	0-0-3	2		100
Total Credit (Practical)					**6**		
Total Semester Credit					**21**		
Total Marks							**300**

COMPUTER NETWORK

UNIVERSITY COURSE CODE: MCA02001

COURSE OUTCOME:

Students will be able to

CO1	Outline the basic concept of networking, types, networking topologies and layered architecture.
CO2	Explain data link layer and MAC sub-layer.
CO3	Demonstrate the network Layer functioning.
CO4	Identify the different types of network devices and their functions within a network.
CO5	Explain the transport layer and application layer operation.
CO6	Design and maintenance of individual networks

Module-I (12 Periods)

Overview of the Internet: introduction to data communication, computer networks, Protocol, Layering Scenario, TCP/IP Protocol Suite: The OSI Model, Internet history, standards and administration; Comparison of the OSI and TCP/IP reference model. **Physical Layer:** data and signals: analog and digital, periodic analog signals, digital signals, transmission impairments, data rate limit, Guided transmission media, unguided transmission media.

Module– II (08 Periods)

Data Link Layer: error detection and correction design issues, CRC codes, Elementary Data Link Layer Protocols, sliding window protocols, noisy and noiseless channels.
Multiple Access Protocols: random access, controlled access, channelization**,** ALOHA, CSMA,

Module – III (06 Periods)
Connecting devices: learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways, definition of multiplexing and types.
Network Layer: Network Layer Design issues, store and forward packet switching, connection less and connection oriented networks-routing algorithms-optimality principle, circuit and packet switching, definition of flooding and multicast**.**

Module – IV (05 Periods)
Routing protocols: Shortest Path, Routing uni-cast Distance Vector Routing, RIP, link state protocols, path vector routing. **Internetworking:** logical addressing, internet protocols**,** IP address, CIDR, IPv4 addressing, IPv6 Protocol addressing, addresses mapping, ICMP, IGMP, ARP, RARP, DHCP.

Module -- V (09 Periods)
Transport Protocols: process to process delivery**,** UDP, TCP, TCP Service Model, TCP Sliding Window, TCP Congestion Control, congestion control and quality of service.
Application Layer- Introduction, providing services, Client server model, Standard client-server application-HTTP, FTP, electronic mail, TELNET, DNS.

Content beyond Syllabus:

1. **Data abstraction and schemes**
2. **Agile process models**
3. **Routing and parsing**
4. **Introduction to cyber security**

Text Books:

1. Behrouz A. Forouzan, "**Data Communications and Networking**", McGraw Hill Publication
2. Andrew S Tanenbaum, "**Computer Networks**", Pearson Education
3. L. L. Peterson and B. S. Davie, "**Computer Networks**", Elsevier.

References:

James F. Kurose, K. W. Ross, "**Computer Networking: A Top-Down Approach Featuring the Internet**", Pearson Education

ANALYSIS AND DESIGN OF ALGORITHM

UNIVERSITY COURSE CODE: M C A 0 2 0 0 2

COURSE OUTCOME:

Students will be able to

CO1	Analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.
CO2	Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms
CO3	Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation
CO4	Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming and develop the dynamic programming algorithms, and analyze it to determine its computational complexity.
CO5	Solve a given model engineering problem model it using graph and write the corresponding algorithm to solve the problems.
CO6	Solve and analyze the NP-Completeness and Reducibility of an algorithm.

Module-I: (8 Periods)

Notion of Algorithm : Growth of functions, Recurrences: The Master method, The Substitution method, The Iteration method, Asymptotic Notations and Basic Efficiency Classes (Use of Big O, θ, etc.) in analysis of algorithms, Mathematical Analysis of few Non-Recursive and Recursive Algorithms.

Module-II: (8 Periods)
Sorting and Searching Techniques : Selection Sort, Bubble Sort, Insertion Sort, Sequential Search, Binary Search, Depth First Search and Breadth First Search, Balanced Search Trees, AVL Trees, Red-Black Trees, Heaps and Heap Sort, Disjoint Set and their Implementation, Divide and Conquer Paradigm of problem solving, Complexity analysis and understanding of Merge Sort, Quick Sort, Binary Search Trees.

Module-III: (8 Periods)
Greedy Techniques: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's and Bellman Ford Algorithm, Huffman Trees, Knapsack problem.
Dynamic Programming Paradigm : Floyd-Warshall Algorithm, Optimal Binary Search trees, Matrix Chain Multiplication Problem, Longest Common Subsequence Problem, 0/1 Knapsack Problem, Maximum Network Flow Problem.

Module-IV: (8 Periods)
String Matching Algorithms: Naive string matching algorithm, The Rabin-Karp Algorithm, string matching with Finite Automata, Knuth Morris Pratt string matching algorithm.
Backtracking: n-Queen's problem, Hamiltonian Circuit problem, Subset-Sum problem, State Space Search Tree for these problems

Module-V: (8 Periods)
Branch and Bound: Travelling Salesman Problem and its State Space Search Tree.
Introduction to Computability: Polynomial-time verification, NP-Completeness and Reducibility, NP-Complete problems.
Approximation Algorithms: Vertex Cover Problem.

Content beyond Syllabus:

1. Examples of NP Hard, NP Complete problems.

Text Books:

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms", PHI Publication.
2. A.V. Aho, J. E. Hopcroft and J.D.Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education.
3. R. S. Salaria, Khanna, "Data Structure & Algorithms", Khanna Book Publishing Co. (P) Ltd.

References:

1. AnanyLevitin, Introduction to the Design and Analysis of Algorithms, Pearson, 3rd Edition.
2. Richard E. Neapolitan,KumarssNaimipour, Foundations of Algorithms using C++ Psuedocode, Second Edition.

Object Oriented Programming Using Java

UNIVERSITY COURSE CODE: MCA02003

COURSE OUTCOME:

Students will be able to

CO1	Recall different Object-oriented programming techniques.
CO2	Describe basic object-oriented features.
CO3	Explain the correlation between object and class.
CO4	Determine inheritance and wrapper classes to achieve code reusability.
CO5	Perform web applications using Applets and collection frame work.
CO6	Design and create GUI based event driven programs using Swing and JavaFX.

Module-I (08 Periods)

JAVA BASICS: Review of Object oriented concepts, History of Java, Java buzzwords, JVM architecture, Data types, Variables, Scope and life time of variables, arrays, operators, control statements, type conversion and casting, simple java program, constructors, methods, Static block, Static Data, Static Method String and String Buffer Classes, Using Java API Document.

Module-II (08 Periods)

INHERITANCE AND POLYMORPHISM: Basic concepts, Types of inheritance, Member access rules, Usage of this and Super key word, Method Overloading, Method overriding, Abstract classes, Dynamic method dispatch, Usage of final keyword. PACKAGES AND INTERFACES: Defining package, Access protection, importing packages, Defining and Implementing interfaces, and Extending interfaces. I / O STREAMS: Concepts of streams, Stream classes- Byte and Character stream, Reading console Input and Writing Console output, File Handling.

Module-III (08 Periods)
EXCEPTION HANDLING: Exception types, Usage of Try, Catch, Throw, Throws and Finally keywords, Built-in Exceptions, Creating own Exception classes. MULTI THREADING: Concepts of Thread, Thread life cycle, creating threads using Thread class and Runnable interface, Synchronization, Thread priorities, Inter Thread communication.

Module-IV (08 Periods)
AWT CONTROLS: The AWT class hierarchy, user interface components- Labels, Button, Text Components, Check Box, Check Box Group, Choice, List Box, Panels – Scroll Pane, Menu, Scroll Bar. Working with Frame class, Colour, Fonts and layout managers. EVENT HANDLING: Events, Event sources, Event Listeners, Event Delegation Model (EDM), Handling Mouse and Keyboard Events, Adapter classes, Inner classes.

Module-V (08 Periods)
SWINGS: Introduction to Swings, Hierarchy of swing components. Containers, Top level containers -JFrame, JWindow, JDialog, JPanel, JButton, JToggleButton, JCheckBox, JRadioButton, JLabel,JTextField, JTextArea, JList, JComboBox, JScrollPane. APPLETS: Life cycle of an Applet, Differences between Applets and Applications, Developing applets, simple applet.

Content beyond Syllabus:

1. Basics of a Web application.
2. Web Container and Web Application Project Set up.
3. To set up Tomcat Container on a machine. Servlets.
4. Session Management.
5. JSPs. Introduction to JSP and need for JSPs.

Text Books:

1. Herbert schildt (2010), The complete reference, 7th edition, Tata Mc graw Hill, New Delhi
2. Programming with Java, E. Balagurusamy, McGraw-Hill Education, 6th Edition.
3. Head First Java, O'rielly publications 2. T. Budd (2009), An Introduction to Object Oriented Programming, 3rd edition, Pearson Education, India.

References:

4. J. Nino, F. A. Hosch (2002), An Introduction to programming and OO design using Java, John Wiley & sons, New Jersey.

5. Y. Daniel Liang (2010), Introduction to Java programming,7th edition, Pearson education, India.

OBJECT ORIENTED ANALYSIS & DESIGN

UNIVERSITY COURSE CODE: M C A 0 2 0 0 4

COURSE OUTCOME:

Students will be able to

CO1	Design OO Application using Patterns.
CO2	Solve real world problems by applying OOAD principle.
CO3	Acquire expertise in Java programming.
CO4	Understanding and documenting the requirements of the system.
CO5	Gathered requirements are analyzed to identify the main objects and their relationships in the problem domain
CO6	Design OO Application using Patterns.

Module-I: (5 Periods)
Introduction:
Basic concepts, abstraction, encapsulation, information hiding, inheritance, dynamic binding, polymorphism, overview of OOAD.

Module-II: (10 Periods)
Unified modeling language (UML):
UML views and diagrams, Use case modeling, actors and use cases, factoring use cases; Class diagrams, class relations, association, inheritance, aggregation/composition, inheritance, dependency; object diagram, Packages, Interaction diagrams, sequence diagrams, fragments, Communication diagram; State diagram, events, guards, composite states, concurrent states, history state; activity diagram, swim lanes, events, messages, object flow, Component diagram, Deployment diagram.

Module-III: (5 Periods)
Object-oriented design process:
Overview of the design process, Domain modeling, identifying objects, boundary objects, control objects, entity objects, CRC cards, CASE support.

Module-IV: (10 Periods)
Basic principles:
SOLID principles, Single Responsibility Principle (SRP), Open-Closed Principle (OCP), Liskov Substitution principle (LSP), Interface segregation Principle (ISP), Dependency Inversion Principle (DIP), Martin's Package metrics, CK metrics, O-O metrics.

Module-V: (10 Periods)
Design Patterns:
Overview of patterns, Architectural, design, and code patterns, GRASP and GoF patterns, Expert, Creator, Law of Demeter, Controller, Singleton, Model View Separation patterns, Observer, MVC, Publish-Subscribe, Singleton, State, Composite, Façade, Decorator, Proxy, Bridge, Strategy, Mediator, Visitor, Iterator, Flyweight, Template, Memento.

Content beyond Syllabus:

1. Forward & Reverse Engineering of all UML diagrams.

Text Books:

1. Grady Booch, Object-Oriented Analysis and Design with Applications (Third Edition), Addison-Wesley.
2. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, (First Edition), Addison-Wesley.
3. Robert C. Martin, UML for Java Programmers, Prentice Hall.

References:

4. RUMBAUGH and BLAHA, Object-Oriented Modeling and Design with UML, Pearson.
5. Bernd Bruegge and, Allen H. Dutoit, Object-Oriented Software Engineering Using UML, Patterns, and Java, Pearson.
6. Bernd Oestereich, Developing Software with UML: Object-Oriented Analysis and Design in Practice, Addison Wesley.

INTERNET AND WEB PROGRAMMING

UNIVERSITY COURSE CODE: MCA02005

COURSE OUTCOME:

Students will be able to

CO1	Understand & Apply practical problems to internet web technology concepts
CO2	Design model problems using standard web technology concepts
CO3	Apply web technology skills in real-world problem solving.
CO4	Understand scripting language concepts for developing client-side applications.
CO5	Understand fundamental tools and technologies for web design
CO6	Design Web Different skills using different tools like PHP.

Module I (8 Periods)

Internet Architecture: Internet overview, evolution of internet. Internet components: Local Area Networks, Access Networks, Core Networks, Routers, Transmission infrastructure, ISPs. TCP/IP model, TCP/IP vs OSI model. HTML: HTML Overview, Structure of HTML Documents, Document Types, HTML Elements and attributes. Anchor Attributes, Image Tag and its attributes, Image and Anchors, Table.

Module II (8 Periods)

Image Map: Attributes, Client Side Image Maps and Server Side Maps.
HTML Layout: Background, colors and text, Tables, Frames, Layers, Page content Division <Div>, <SPAN>. CSS: Style Sheet Basic, Properties, Positioning with Style Sheet.
Forms: <FORM> Elements, Form controls. Dynamic HTML.

Module III (8 Periods)

Java Script: Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security. Operators: Assignment Operators, Comparison Operators, Arithmetic Operators, Increment, Decrement, Unary Negation, Logical Operators, String Operators, Special Operators, Conditional operator, Comma operator, delete, new, this, void.
Statements: Break, comment, continue, delete, do … while, export, for, for…in, function, if…else, import, labeled, return, switch, var, while.

Module IV (8 Periods)

JavaScript (Properties and Methods of Each) :Array, Boolean, Date, Function, Math, Number, Object, String, regExp. Document and its associated objects, document, Link, Area, Anchor, Image, Applet, Layer.
Events and Event Handlers: General Information about Events, Defining Event Handlers, event.

Module V (8 Periods)

Server Side Programming: Common Gateway Interface (CGI), Active Server Pages.
Internet applications: FTP, Telnet, Email, Chat. World Wide Web: HTTP protocol. Search Engines. E-commerce and security issues including symmetric and asymmetric key, encryption and digital signature, and authentication. Emerging trends, Internet telephony, and virtual reality over the web, etc. Intranet and extranet, firewall.

Content beyond Syllabus:

1. Develop server side programming techniques to solve real time application
2. Apply JSP concept

Text Books:

1. Computer Networking: A Top-Down Approach Featuring the Internet by Kurose and Ross, Pearson.
2. Web Design the Complete Reference by Thomas Powell, Tata McGraw Hill.
3. HTML The Complete Reference by Thomas Powell, Tata McGraw Hill.

References:

4. JavaScript the Complete Reference, Second Edition by Thomas Powell, Fritz Schneider.Tata McGraw Hill.

JAVA AND PYTHON PROGRAMMING LAB

UNIVERSITY COURSE CODE: MCA02006

COURSE OUTCOME:

Students will be able to

CO1	Develop database modeling for a problem.
CO2	Design a database using normalizations
CO3	Implement a database query language.
CO4	Develop GUI using front end tool
CO5	Develop a connection between frontend and database
CO6	Implement a Data Manipulation Language

LIST OF EXPERIMENTS

Java Programming

1. Write a program in Java to find the set of prime numbers from 1 to 100.
2. Write a program to compare two objects. Create two objects representing two complexnumber and find the larger one.
3. Write a Java Program to convert a Number to Word.
4. Write a Java Program to copy all elements of one array into another array
5. Write a Java Program to sort the elements of an array in ascending order
6. Write a Java Program to find the frequency of odd & even numbers in the given matrix
7. Write a Java Program to determine whether a given string is palindrome

8. Write a Java program to draw a pattern such

2 4 3 6 9 4 8 12 16	000*000* 0*00*00*0 00*0*0*00 000***000

9. Write a Java program to convert Decimal to Binary in Java

10. Write a program to add two times given in hour minutes and seconds using class and object.
11. Write a Java program to find the combination c(n,r) by inheriting from a class that computes the factorial of a number.
12. Write a Java program to find the area of different geometrical shapes using polymorphism.
13. Write a Java program to create a user defined package that finds the largest among an array of *n* numbers. Use this package to sort an array of *n* numbers using insertion/selection sort.
14. Create three threads and print 1 to 10 in each thread.
15. Write a Java program to illustrate the concept of some exceptions such as divide by zero or array index out of bound etc.

Python Programming

1. Write a Program to read and print values of variables of different data types.
2. Write a program to perform addition, subtraction, multiplication, division and modulo division on two integers.
3. Write a program to input two numbers and check whether they are equal or not.
4. Write a program that prompts user to enter a character (O, A, B, C, F). Then using if-elseif-else construct display Outstanding, Very Good, Good, Average and Fail respectively.
5. Write a program to print Fibonacci series using recursion.
6. Write a program that prints absolute value, square root and cube root of a number. (Import math package).
7. Write a program that finds the greatest of three given numbers using functions. Pass three arguments.
8. Write a program to get a string made of the first 2 and last 2 characters from a given string. If the string length is less than 2, return empty string.
9. Write a program that fetches data from a specified url and writes it in a file.
10. Write a program to find the resolution of an image

COMPUTER NETWORKS LAB

UNIVERSITY COURSE CODE: MCA02007

COURSE OUTCOME:

Students will be able to

CO1	Explain OSI Reference Model and in particular have a good knowledge of Layers 1-3
CO2	Design the working knowledge of datagram and internet socket programming
CO3	Design and test simple programs to implement networking concepts using Java
CO4	Design simple data transmission using networking concepts and implement
CO5	Compare and analyze different existing protocols
CO6	Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies

LIST OF EXPERIMENTS

Simulate the following using any programming language

1. Error detection in a packet using Checksum
2. Simplex stop-and-wait protocol with positive acknowledgement and retransmission
3. Error detection using CRC-CCITT (16-bits)
4. Token-Bus medium access scheme
5. Selective repeat sliding window protocol
6. Congestion control using leaky bucket algorithm.
7. Find all pair shortest path between vertices using bellman-ford algorithm
8. Client/Server message passing, where a client1 send a character to a server, which on receiving the character increment it to the next letter in the alphabet, and sends the character to client2. The client2 on receiving the value from server, print it and all process terminates.
9. Client/Server message passing, where a client1 send a message that is structure containing values of type character, integer and float to the server. The server should print the message using the format “char value %c integer value %d float value %f” before passing it to the next client. The server should change the value of each element of the structure before passing it to client2. The client2 should print the structure values it receives from the server using the above format.

ALGORITHM DESIGN LAB

UNIVERSITY COURSE CODE: MCA02008

COURSE OUTCOME:

Students will be able to

CO1	Discuss different computational models for example divide and conquer,order notation ( ), various Complexity measures to analyze the performance of different algorithms
CO2	Understand the difference between the lower and upper bounds of various problems and their Importance in deciding the optimality of an algorithm
CO3	Analyze various techniques for efficient algorithm design (divide and conquer, greedy, and Dynamic programming algorithms) and able to apply them while designing algorithms
CO4	Apply different designing methods development of algorithms using greedy method (application)
CO5	Apply backtracking and branch and bound techniques to deal with some hard problems
CO6	Know the concepts of tractable and intractable problems, classes of P,NP,NP-HARD and NP-Complete problems.

LIST OF EXPERIMENTS:

1. Implementation of Stack and Queue – Operations and Applications.
2. Implementation of different searching algorithms.
3. Implementation of different sorting algorithms.
4. Problem solving using Divide and Conquer technique.
5. Problem solving using Dynamic Programming technique.
6. Problem solving using Greedy technique.
7. Problem solving using Backtracking technique.
8. Problem solving using disjoint-set data structure operations.
9. Problem solving using Branch and Bound technique.
10. Problem solving for the Maximum Flow problem.
11. Implementation of Graph Traversal algorithms – Breadth-First-Search (BFS) and Depth-First-Search (DFS).
12. Implementation of Minimum Spanning Tree construction algorithms – Kruskal and Prim.
13. Implementation of different String-Matching algorithms.
14. Problem solving for the Shortest Path problem using different algorithms.
15. Problem solving using Approximation algorithms.

Third Semester							
Theory							
Sl. No.	Category	Course Code	Course Title	L-T-P	Credit	University Marks	Internal Evaluation
1	PC	MCA03001	Software Engineering	3-0-0	3	100	50
2	PC	MCA03002	Compiler Design	3-0-0	3	100	50
3	NPTEL-MOOC	MCA03003 (NM-)	Elective-I (To be opted from NPTEL MOOC Pool)		3	-	-
4	NPTEL-MOOC	MCA03004 (NM-)	Elective-II (To be opted from NPTEL MOOC Pool)		3	-	-
5	NPTEL-MOOC	MCA03005 (NM-)	Elective-III (To be opted from NPTEL MOOC Pool)		3	-	-
Total Credit (Theory)					**15**		
Total Marks						**200**	**100**
Practical							
1	PC	MCA03006	Software Engineering Lab	0-0-3	2		100
2	PC	MCA03007	Seminar and Technical Writing	0-0-3	2		100
3	PC	MCA03008	Web Programming Lab	0-0-3	2		100
Total Credit (Practical)					**6**		
Total Semester Credit					**21**		
Total Marks							**300**

SOFTWARE ENGINEERING

UNIVERSITY COURSE CODE: MCA03001

COURSE OUTCOME:

Students will be able to

CO1	Acquire strong fundamental knowledge in science, mathematics, fundamentals of computer science, software engineering and multidisciplinary engineering to begin in practice as a software engineer.
CO2	Design applicable solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal and economic concerns.
CO3	Create quality software products by possessing the leadership skills as an individual or contributing to the team development and demonstrating effective and modern working strategies by applying both communication and negotiation management skill.
CO4	Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development
CO5	Develop an awareness of the role and responsibilities of the professional software engineer; .
CO6	Acquire skills to think about problems and their solutions using appropriate methods of analysis and design; .

Module I:

Software development life cycle and Project Management: Software development life cycle (SDLC) models such as Waterfall model, Iterative waterfall model, Prototyping model, Evolutionary model, Spiral model, V model, RAD, Agile models etc., software project management, project planning, metrics for project size estimation such as LOC and FP, project estimation, COCOMO, Halstead's software science, Staffing level estimation, project scheduling, staffing, Organization and team structure, risk management, configuration management. [11hours]

Module II:

Requirements analysis and specification: Requirements gathering and analysis, software requirements specification, formal systems specification. [3 hours]

Module III:

Software Design: Outcome of a design process, cohesion and coupling, layered arrangement of modules, approaches to software design, function-oriented software design: overview of SA/SD methodology, structured analysis, DFDs, Data Dictionary, structured design, detailed design, object-oriented software design: UML diagrams such as use case diagram, class diagram, object diagram, sequence diagram, communication diagram, state chart diagram, activity diagram, etc., unified process, OOD goodness criteria. [11 hours]

Module IV:

Coding and Testing: Coding standards and guidelines, code review, software documentation, unit testing, black-box testing, white-box testing, debugging, integration testing, system testing, performance testing, regression testing. [8 hours]

Module V : Software reliability and Quality management: Software reliability, Statistical testing, software quality, software quality management system,ISO 9000, SEI CMM, PSP, Six sigma, CASE Tools, Software maintenance, Software reuse. [7 hours]

Content beyond Syllabus:

1. Software quality factors
2. Debug Testing

Text Books:

1. R. Mall, Fundamentals of Software Engineering, 5th Edition, PHI Learning, 2018.
2. R. S. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill Publications, 2015.
3. I. Sommerville, Software Engineering, Pearson Education, 2015.
4. K. K. Aggarwal and Yogesh Singh, "Software Engineering", New Age International Publishing, 2007.

Reference Book:

5. Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa Publication, 2019.
6. A. Behferooz and F. J. Hudson, Software Engineering Fundamentals, Oxford University Press, 2014.
7. James Peter, W. Pedrycz, "Software Engineering: An Engineering Approach", John Wiley & Sons, 2000.

COMPILER DESIGN

UNIVERSITY COURSE CODE: MCA03002

COURSE OUTCOME:

Students will be able to

CO1	Define various phases of compiler, code optimization techniques and machine code generation.
CO2	Classify top down & bottom-up parsing.
CO3	Demonstrate DAG for intermediate code generation.
CO4	Analyze the knowledge of parser by parsing LL parser and LR parser.
CO5	Analyze & Design Run time environments and Syntax directed translations.
CO6	Implementing code optimization by removing redundant and unreachable codes.

Module- I: (8 Periods)

Compiler Structure: Model of compilation, various phases of a compiler. Lexical analysis: Interface with input parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, input buffering. Specification of tokens. Regular grammar & language definition.

Module- II: (12 Periods)

Syntax Analysis: Grammar, Parsing, ambiguity, top down parsing, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing LL(1) grammar, Nor LL(1) grammar, Bottom up parsing, operator precedence grammars, LR parsers (SLR, CLR, LALR).

Module- III: (10 Periods)

Syntax directed definitions: Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions. Type checking: type: type system, type expressions, structural and name equivalence of types, type conversion. Run time system: storage organization, activation tree, activation record, parameter passing symbol table, dynamic storage allocation.

Module- IV: (10 Periods)

Intermediate code generation: intermediate code representation techniques. Intermediate Code generation for control flow, function call, Boolean expressions and procedure calls. Code optimization: source of optimizations, optimization of basic blocks, loops, global dataflow analysis, solution to iterative dataflow equations, code improving transformations, dealing with aliases, data flow analysis of structured flow graphs.

Module- V: (10 Periods)

Code generation and instruction selection: Issues, basic blocks and flow graphs, register allocation, code generation, DAG representation of programs, code generation from DAGS, peep hole optimization. Symbol table management: Data structure for symbol table organization. Error Handling and recovery.

Content beyond Syllabus:

1. ANTLR
2. JAVACC

Text Books:

1. K. C. Louden, “Compiler Construction, Principle and Practice”, Cengage Publication

2. Alfred V. Aho, Ravi Sethi, and Ullman, “Compilers Principles, Techniques and Tools”,Pearson Publication

3. V.Raghvan, “Principles of Compiler Design”, TMH Publication

4. Levine, Mason and Brown, “Lex & Yacc”, O’ Reilly Publication

3rd Semester	**MCA03003**	**Elective-I (To be opted from NPTEL MOOC Pool)**	**L-T-P 3-0-0**	**3 CREDITS**

3rd Semester	**MCA03004**	**Elective-II (To be opted from NPTEL MOOC Pool)**	**L-T-P 3-0-0**	**3 CREDITS**

3rd Semester	**MCA03005**	**Elective-III (To be opted from NPTEL MOOC Pool)**	**L-T-P 3-0-0**	**3 CREDITS**

NPTEL MOOC Pool(For Elective-I, Elective-II and Elective-III)

(Student must choose a Course of 8 weeks or more duration and must submit the relevant certificate from NPTEL to the University through the NPTEL Local Chapter before completion of the 4th Semester for the required credit transfer. No University examinations will be conducted for these subjects. Faculty mentors are to be assigned for guiding and monitoring these students through the corresponding NPTEL local chapters)

Subject Code	Subject Name
NM-1	Artificial Intelligence
NM-2	Soft Computing
NM-3	Computer Network security
NM-4	Information System Design
NM-5	Real-time System
NM-6	Mobile Computing
NM-7	Introduction to Data Science
NM-8	Machine Learning
NM-9	Internet-of-Things
NM-10	Big-Data Analytics
NM-11	Cyber Law and Security
NM-12	Intellectual Property Rights
NM-13	Embedded System
NM-14	Management Information System
NM-15	Digital Image Processing
NM-16	Data Mining
NM-17	Advanced Computer Networks
NM-18	Distributed Operating System
NM-19	Cloud Computing
NM-20	Simulation and Modeling
NM-21	Wireless Sensor Networks
NM-22	Software Project management
NM-23	Advance Database Management Systems

NM-24	Data Analytics
NM-25	Advanced Computer Architecture
NM-26	Intelligence Data Analysis
NM-27	Deep Learning
NM-28	E-Commerce and ERP
NM-29	Computer Graphics and Multimedia
NM-30	Computer Based Optimization techniques

SOFTWARE ENGINEERING LAB

UNIVERSITY COURSE CODE: MCA03006

COURSE OUTCOME:

Students will be able to

CO1	Develop SRS document, design documents such as ER Diagrams, DFDs, UML Diagrams etc. for a given software project.
CO2	Develop efficient codes for a given software project using appropriate coding standards and guidelines and test the developed code using different tools.
CO3	Implement different software project management techniques such as FP, COCOMO, CPM, PERT etc.
CO4	Know the use of different computer aided software engineering (CASE) tools in the development, maintenance and reuse of software systems.
CO5	Estimate the size of a given software using Function Point Metric.
CO6	Perform various testing operations using the available testing tools for a given system.

List of Experiments

1. Prepare the SRS document for a given problem, such as the below mentioned problems. You should identify the appropriate requirements for the given problem; Draw the E-R Diagram using any available tool, Draw the DFD for the given problem using any available tool, Draw the Use Case diagram, Domain Models, and Class Diagram, Sequence Diagrams and Collaboration Diagrams for each Use Case, State Chart Diagram and Activity Diagram, (if necessary) using any available tool; Develop the corresponding software using any programming language such as Java,Python, etc. with an interactive GUI and appropriate Database.

    a) Develop software to automate the bookkeeping activities of a 5 star hotel
    b) The local newspaper and magazine delivery agency wants to automate the various clericalactivities associated with its business. Develop a software for this.
    c) A small automobile spare parts shop sells the spare parts for vehicles of several makes and models. Each spare part is typically manufactured by several small industries. To streamline the sales and supply ordering, the shop owner wants to automate the activities associated withhis business. Develop a software for this.
    d) Develop a software for the automation of the dispensary of your college.
    e) Develop software for automating various activities of the Estate Office of your college.
    f) Develop a word processing software with some limited number of facilities such as makingbold italics, underline, cut, copy and paste etc.
    g) Develop a graphics editor software package, using which one can create / modify severalcommon types of graphics entities.
    h) Develop a software for automating various activities of the departmental offices of yourcollege.

2. Estimate the size of a given software using Function Point Metric.
3. Write a C function for searching an integer value from a large sorted sequence of integer values stored in array of size 100, using the binary search method. Build the control flow graph (CFG) of this function using any compiler writing tool. Write a program in Java to determine its cyclomatic complexity. Identify the linearly independent paths and generate the test cases using path coverage based strategy.
4. To perform various testing operations using the available testing tools for a given system.
5. Write a program in Java to determine the number of defects still remaining after testing, usingerror seeding methodology.
6. Draw the GANT chart for a given software project using any available tool such Gantt Project.
7. Draw the network diagram, find out the critical path and critical activities, and calculate the project duration for a given problem using CPM. You may use any availabletool for this such as Gantt project, ProjectLibre etc.
8. Draw the network diagram, find out the critical path and critical activities, and calculate the project duration for a given problem using PERT. You may use any availabletool for this such as Gantt project, ProjectLibre etc.

WEB PROGRAMMING LAB

UNIVERSITY COURSE CODE: MCA03008

COURSE OUTCOME:

Students will be able to

CO1	Understand the components of web technology
CO2	Develop web pages using CSS
CO3	Design websites using PHP
CO4	Develop different information interchange formats using XML
CO5	Apply and implement client/server application
CO6	Implement web page using server side scripting

List of Experiments:

1. Web design environment : HTML elements coding and testing
2. Implementation of frames and frame elements
3. Write a JavaScript to design a simple calculator to perform the following operations: sum,product, difference and quotient.
4. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 andoutputs HTML text that displays the resulting values in an HTML table format.
5. Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.
6. Develop and demonstrate a HTML file that includes JavaScript script that uses functions forthe following problems: a. Parameter: A string Output: The position in the string of the left-most vowel b. Parameter: A number Output: The number with its digits in the reverse order
7. Design an XML document to store information about a student in an engineering college affiliated to BPUT. The information must include USN, Name, Name of the College, Brach,Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheetand use it to display the document.

8. Write a PHP program to keep track of the number of visitors visiting the web page and todisplay this count of visitors, with proper headings.
9. Write a PHP program to display a digital clock which displays the current time of the server.
10. Write the PHP programs to do the following:
    a. Implement simple calculator operations.
    b. Find the transpose of a matrix.
    c. Multiplication of two matrices.
    d. Addition of two matrices.
11. Write a PHP program named states that declares variable states with value "MississippiAlabama Texas Massachusetts Kansas". write a PHP program that does the following:
    a. Search for a word in variable states that ends in xas. Store this word in element 0 of a listnamed states List.
    b. Search for a word in states that begins with k and ends in s. Perform a case insensitivecomparison. [Note: Passing re.Ias a second parameter to method compile performs a case insensitive comparison.] Store this word in element1 of states List.
    c. Search for a word in states that begins with M and ends in s. Store this word in element 2of the list.
    d. Search for a word in states that ends in a. Store this word in element 3 of the list
12. Write a PHP program to sort the student records which are stored in the database using selection sort. 11. Web Technology Lab with Mini Project

Fourth Semester							
Theory							
Sl. No.	Category	Course Code	Course Title	L-T-P	Credit	University Marks	Internal Evaluation
Practical							
1	PC	MCA04001	Comprehensive Viva-Voice	0-0-2	2		100
2	PC	MCA04002	Internship/ Major Project	0-0-8	15		500
Total Credit (Practical)					**17**		
Total Semester Credit					**17**		
Total Marks							**600**