FOR ADMISSION BATCH 2024-25
MASTER IN COMPUTER APPLICATION
FIRST YEAR (FIRST SEMESTER)

Sl. No.	Category	Course Code	Course	Contact Hrs. L-T-P	Credit	University Marks	Internal Evaluation
Subject							
1.	BS	MCBS1001	Discrete Mathematics	3-0-0	3	100	50
2.	PC	MCPC1001	Digital Logic Design	3-0-0	3	100	50
3.	PC	MCPC1002	Computer Networks	3-0-0	3	100	50
4.	PC	MCPC1003	Programming for Problem Solving	3-0-0	3	100	50
5.	PC	MCPC1004	Database Management Systems	3-0-0	3	100	50
6.	HS	MCHS1001	Communicative English	2-0-0	2	100	50
7.	PC	MCPC1201	Computer Networks Lab.	0-0-3	1.5	-	100
8.	PC	MCPC1202	C Programming Lab	0-0-3	1.5	-	100
9.	PC	MCPC1203	Database Management Systems Lab	0-0-3	1.5	-	100
10.	HS	MCHS1201	Language Lab	0-0-3	1.5	-	100
Total				**17-0-12**	**23**	**600**	**700**

FIRST YEAR (SECOND SEMESTER)

Sl. No.	Category	Course Code	Course	Contact Hrs. L-T-P	Credit	University Marks	Internal Evaluation
Subject							
1.	PC	MCPC1005	Object-Oriented Programming using JAVA	3-0-0	3	100	50
2.	PC	MCPC1006	Software Engineering	3-0-0	3	100	50
3.	PC	MCPC1007	Data Structures	3-0-0	3	100	50
4.	PC	MCPC1008	Computer Organization and Architecture	3-0-0	3	100	50
5.	PC	MCPC1009	Theory of Computation	3-0-0	3	100	50
6.	HS	MCHS1002	Universal Human Values & Professional Ethics	2-0-0	2	100	50
7.	PC	MCPC1204	Object-Oriented Programming Lab	0-0-3	1.5	-	100
8.	PC	MCPC1205	Software Engineering Lab	0-0-3	1.5	-	100
9.	PC	MCPC1206	Data Structures Lab	0-0-3	1.5	-	100
10.	PC	MCPC1207	Programming in Python Lab	0-0-3	1.5	-	100
Total				**17-0-12**	**23**	**600**	**700**

Note: Click here to view/download the syllabus of the subjects.

MCBS1001 DISCRETE MATHEMATICS (3-0-0)

Course Objectives:

- To learn the mathematical foundations required for computer science.
- This course will help in understanding other courses in computer science.

Learning Outcomes:

Upon completion of this course, students will be able to:

CO1 : Define & describe various logical connectives and expressions along with rules of inferences.

CO2 : Apply various methods of proofs and proof strategies.

CO3 :learn the concepts of function and develop the various algorithms and its complexity.

CO4 : Model counting techniques using recurrence relations & generating functions for applications.

CO5 : Develop the concepts and applications of graphs in various computer science problems

UNIT-1:

Logic and Proofs: Propositional logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs. Sets: Venn Diagrams, Subsets, The size of a set, Power Sets, Cartesian Products, Set Operations.

UNIT-2:

Functions: One-to-One and Onto Functions, Inverse Functions and Compositions of Functions Partial Functions. Sequences and Summations. Algorithms, Searching Algorithms: Linear Search, Binary Search, Sorting: Bubble Sort, Insertion Sort, The Growth of Functions, Complexity of Algorithms.

UNIT-3:

Counting: The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, Recurrence Relations.
Relations: Relations and their Properties, n-ary Relations and their Applications, Representing Relations, Closure of Relations, Equivalence Relations, Partial Orderings.

UNIT-4:

Graphs: Graph Terminology and Special Types of Graphs, Bipartite Graphs, Representing Graphs: Isomorphism of Graphs, Euler and Hamilton Paths, Shortest Path Problems: Dijkstra's Algorithm, Traveling Salesperson Problem, Planar Graphs, Graph Coloring. Trees: Tree Traversal, Minimum Spanning Trees.

Text Books:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Mc Graw Hills International Seventh Edition.
2. C. L. Liu, "Elements of Discrete Mathematics", McGraw Hills International Second Edition.

Reference Books:

1. Elements of Discrete Mathematics by C. L. Liu and D.P. Mohapatra, TMH, 2012
2. J. P Tremblay, R. Manohar, "Discrete Mathematical Structures with Applications to Computer Science", TMH, 1997.

MCPC1001 DIGITAL LOGIC DESIGN (3-0-0)

Course Objectives:

1. To introduce the fundamental concepts of digital logic and Boolean algebra.
2. To develop and understanding of combinational and sequential logic circuits.
3. To explore advanced topics such as memory elements, state machines, and programmable logic devices.

Course Outcomes: Upon successful completion of this course, students should be able to:

CO1 : Analyze and design combinational logic circuits using Boolean algebra and Karnaugh maps.
CO2 : Design and implement sequential logic circuits, including flip-flops, counters, and registers.
CO3 : Apply knowledge of digital logic to solve real-world engineering problems.

Unit 1:

Binary Systems: Digital Computers and Digital Systems, Binary Numbers, Number Base Conversions, Octal and Hexadecimal Numbers, Complements, Signed Binary Numbers, Boolean Algebra and Logic Gates: Boolean functions, Logic Operators, digital Logic Gates, Simplification of Boolean functions: Two and Three Variable Maps, Four Variable Map, Five Variable Map, Product of Sums Simplification, NAND and NOR Implementation, Don't Care Conditions.

Unit 2:

Combinational Logic: Design Procedure, Adders, Subtractors, Code Conversion, Analysis Procedure, Multilevel NAND Circuits, Multilevel NOR Circuits, Exclusive OR Functions, Binary Adder and Subtractor, Decimal Adder, Magnitude Comparator, Decoders and Encoders, Multiplexers, Programmable Logic Array (PLA), Programmable Array Logic (PAL).

Unit 3:

Flip-Flops: RS Flip-Flop, D Flip-Flop, JK and T Flip-Flops, Triggering of Flip-Flops, Analysis of Clocked Sequential Circuits, State Reduction and Assignment, Design Procedure, Design of Counters, Registers, Shift Register, Ripple Counters, Synchronous Counters, Timing Sequences, Random-Access Memory (RAM)

Unit 4:

Semiconductor RAM Memories: Internal Organization of Memory Chips, Static Memories, Dynamic RAMs, Read-only Memories: ROM, PROM, EPROM, EEPROM, Flash Memory, Direct Memory Access, Memory Hierarchy, Cache Memory, Virtual Memory, Secondary

Storage: Magnetic Hard Disks, Optical Disks, Magnetic Tape Systems Memory elements: SRAM, DRAM, ROM, Programmable logic arrays (PLAs) and field-programmable gate arrays (FPGAs), Introduction to hardware description languages (HDLs) such as Verilog or VHDL, Introduction to digital simulation tools

Text Books:

1. "Digital Design" by M. Morris Mano and Michael D. Ciletti
2. "Fundamentals of Digital Logic with Verilog Design" by Stephen Brown and Zvonko Vranesic
3. “Computer Organisation and Embedded Systems” by Carl Hamacher, Z Vranesic, S Zaky and N Manjikian

Reference Books:

1. "Digital Systems: Principles and Applications" by Ronald J. Tocci, Neal S. Widmer, and Greg Moss
2. "Introduction to Logic Design" by Alan B. Marcovitz

MCPC1002 COMPUTER NETWORKS (3-0-0)

Objective :

1. Introduce students to the architecture, standards, and protocols of computer networks.
2. Provide an understanding of the functionalities of various network layers, including physical, data link, network, transport, and application layers.
3. Discuss the principles of routing, addressing, and internetworking in modern network environments.
4. Familiarize students with network applications, standard protocols, and techniques for ensuring quality of service and congestion control.

Module-I

Overview of the Internet: introduction to data communication, network application, Network hardware, Protocol, Layering Scenario, reference models: The OSI Model, TCP/IP 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, Wireless transmission, mobile telephone system.

Module–II

Data Link Layer: Design issues, error detection and correction design issues, elementary data link protocols, CRC codes, sliding window protocols, HDLC, the data link layer in the internet. Elementary Data Link Layer Protocols, sliding window protocols, noisy and noiseless channels.

THE MEDIUM ACCESS SUBLAYER: Channel allocations problem, multiple access protocols, Ethernet, Data Link Layer switching, Wireless LAN, Broadband Wireless, Bluetooth.

Module–III

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, connectionless and connection oriented networks-routing algorithms-optimality principle, circuit and packet switching, definition of flooding and multicast.

Module– IV

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

Transport Protocols: process to process delivery, UDP, TCP, TCP Sliding Window, TCP Congestion Control, congestion control and quality of service.

Application Layer-World Wide Web, Standard client-server application-HTTP, FTP, electronic mail, TELNET, DNS.

Course Outcome :

Upon successful completion of this course, students will be able to:

1. Explain the architecture and functioning of different network layers and their associated protocols.
2. Compare the OSI and TCP/IP reference models and understand their application in real-world networks.
3. Implement and troubleshoot data link layer protocols and error detection/correction methods.
4. Design and manage network systems using appropriate hardware and software tools, including IP addressing and routing protocols.
5. Utilize and manage network applications and protocols such as HTTP, FTP, email, TELNET, and DNS effectively.

Text Books :

1. "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross.
2. "Data Communications and Networking" by Behrouz A. Forouzan.

References :

1. Computer networks by Tanenbaum, A.S., Pearson Education India.
2. Computer Networks by Bhushan Trivedi, Oxford University Press

MCPC1003 PROGRAMMING FOR PROBLEM SOLVING (3-0-0)

Course Objectives:

- To provide an understanding of basic programming concepts using the C programming language.
- To develop problem-solving skills using C programming constructs.
- To introduce students to algorithmic thinking and program design techniques.
- To enable students to write, compile, and debug programs in C.

Course Outcomes (CO):

CO1: Understand the fundamental concepts of programming using the C language.
CO2: Develop problem-solving skills through the application of programming constructs in C.
CO3: Design and implement functions and algorithms to solve complex problems.
CO4: Demonstrate proficiency in using pointers, arrays, and structures in C programming.
CO5: Apply error handling and debugging techniques to identify and resolve programming errors.
CO6: Utilize file handling mechanisms in C for input/output operations.
CO7: Appreciate the importance of data structures and their implementation in C.

Unit 1: Introduction to C Programming

Introduction to Problem Solving through programs, Flowcharts/Pseudo codes, the compilation process, Syntax and Semantic errors, Variables and Data Types , Arithmetic expressions, Relational Operations, Logical expressions; Conditional Branching and Iterative Loops.

Unit 2: Functions and Arrays

Introduction to Functions, Function Prototypes and Declarations, Parameter Passing in Functions, Recursion, Arrays: 2-D arrays, Character Arrays and Strings.

Unit 3: Pointers and Structures

Introduction to Pointers, Pointer Arithmetic, Dynamic Memory Allocation, Structures and Unions
File Handling in C, Self-Referential Structures and Introduction to Lists.

Unit 4: Advanced Concepts in C

Preprocessor Directives, Command Line Arguments, Bitwise Operators, Error Handling and Debugging Techniques, Introduction to Data Structures in C.

Textbooks:

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill

Reference Books:

1. "C Programming: A Modern Approach" by K.N. King
2. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
3. "Let Us C" by Yashavant Kanetkar
4. "Programming in C" by Stephen G. Kochan

MCPC1004 DATABASE MANAGEMENT SYSTEMS (3-0-0)

Course Objective:

This course provides fundamental and practical knowledge on database concepts by means of organizing the information, storing and retrieve the information in an efficient and a flexible way from a well-structured relational model. This course ensures that every student will gain experience in creating data models and database design and be able to do the followings.

Focus the role of a database management system in an organization and construct ER Diagram.

Demonstrate basic database concepts, including the structure and operation of the relational data model and basic database queries using SQL.

Applying advanced database queries using Structured Query Language (SQL).

Evaluating logical database design principles and database normalization.

Demonstrate the concept of a database transaction, concurrency control, and data object locking and protocols.

Course Outcomes:

After successful completion of the course the student will be able to:
CO1: Understand database design principles.

CO2: Apply data Modelling using E-R diagrams.

CO3: Create refined data models using normalization.

CO4: Build database queries using Structured Query Language.

CO5: Understand the transaction management and concurrency control.

UNIT – 1
Introduction to DBMS: File system vs. DBMS, advantages of DBMS, storage data, queries, DBMS structure, Types of Databases – Hierarchical, Network, Relational, Key-Value, Object Oriented, XML DB Overview of File Structures in database, 3-schema architecture of DBMS, data independence, EF Codd Rule.

UNIT – 2
Data base Design: Data models, the importance of data models. E-R model: Entities, attributes and entity sets, relationship and relationship set, mapping cardinalities, keys, features of ER model, conceptual database design with ER model.
Relational model: Integrity constraints over relations and enforcement, querying relation data, logical database design, views, destroying/altering tables and views, Relational algebra, Extended relational algebra Operations.

UNIT- 3

Schema Refinement and Normal Forms: Introduction to Schema Refinement, Functional Dependencies, Reasoning about Functional Dependencies. Normal Forms, Properties of Decomposition, Normalization, different types of dependencies.

UNIT – 4

Basic SQL: Introduction to SQL, Basic SQL Queries: DML, DDL, DCL, and TCL

Structured Query Language (SQL): Select Commands, Union, Intersection, Except, Nested Queries, Aggregate Operators, Null values, Relational set operators, SQL join operators

Relational Algebra (RA): Selection, Projection, Set operations, joins

Relational Calculus (TRC, DRC): Tuple Relational Calculus, Domain Relational Calculus PL/SQL, Assertions, Triggers. **Introduction to Transaction Management**: ACID properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control. Concurrency Control: 2PL, Serializability and Recoverability, Introduction to Lock Management, Lock Conversions, Dealing with Deadlocks, Concurrency control without locking. Crash Recovery: Aries, Recovering from a System Crash.

Advanced Database: OODB, WEB based DB, Data warehousing and Data mining.

Textbooks:

1. H.F. Korth, A. Silverschatz, Abraham," Database system concepts", Tata McGraw Hill Publication, 6e, 2011
2. Raghu Ramakrishna and Johannes Gehrke, Database Management Systems, McGraw-Hill, 3e, 2014

References:

1. D. Ullman, Principles of Database and Knowledge – Base Systems, Vol. 1, 1/e, Computer Science Press, 1990.
2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Education, 7e, 2016.
3. Er. Rajiv chopra, "Database management systems, A Practical Approach", S. Chand Publishing

MCHS1001 COMMUNICATIVE ENGLISH (2-0-0)

Course Objectives:

This course is designed to enhance the communication skills of MCA students, focusing on the specific needs of computer science professionals. The syllabus aims to develop proficiency in English for academic, professional, and everyday use.

Course Outcomes:

CO1: Students will be able to articulate the basic principles and processes of communication, identify and overcome common barriers, and distinguish between verbal and non-verbal communication methods.

CO2: Students will demonstrate improved listening skills through active listening techniques, effective comprehension, and the ability to engage in clear and confident public speaking, group discussions, and role plays.

CO3: Students will develop proficiency in writing professional documents including emails, memos, business letters, and technical reports, ensuring proper format, etiquette, and avoidance of plagiarism.

CO4: Students will be capable of preparing and delivering effective presentations using appropriate visual aids and tools, while also demonstrating a strong grasp of English grammar including state and event verbs, tense and aspect, and subject-verb agreement.

CO5: Students will understand the dynamics of interpersonal communication, the importance of workplace ethics, and cross-cultural communication. They will also learn to effectively communicate within teams, understand roles and responsibilities, and utilize collaborative tools and technologies.

CO6: Students will enhance their reading comprehension and critical analysis skills for both technical and non-technical texts, expand their vocabulary with strategies for learning new words and technical terms, and develop skills for writing effective blogs, social media posts, and website content.

UNIT 1: BASICS OF COMMUNICATION

1. Introduction to Communication: Definition and Process; Types of Communication: Verbal and Non-verbal; Barriers to Effective Communication

2. Listening Skills: Active Listening Techniques; Barriers to Effective Listening; Listening Comprehension Exercises

3. Speaking Skills: Basics of Pronunciation and Intonation; Public Speaking: Techniques and Practice; Group Discussions and Role Plays

UNIT 2: PROFESSIONAL COMMUNICATION

1. Business Writing: Email Writing: Format and Etiquette; Writing Memos and Notices; Business Letters: Inquiry, Complaint, and Job Application Letters; Writing Technical Reports; Avoiding Plagiarism

2. Presentation Skills: Preparing Effective Presentations; Visual Aids: Use of PowerPoint and Other Tools; Delivering Presentations with Confidence

3. Basics of English Grammar: State and Event Verbs; Tense and Aspect; Subject-Verb Agreement

UNIT 3: INTERPERSONAL SKILLS

1. Interpersonal Communication: Building Relationships through Communication; Importance of Ethics at the Workplace; Cross-Cultural Communication

2. Teamwork and Collaboration: Effective Team Communication; Roles and Responsibilities in a Team; Collaborative Tools and Technologies

3. Interview Skills: Preparing for an Interview; Common Interview Questions and Answers; Mock Interviews and Feedback

UNIT 4: ENHANCING LANGUAGE SKILLS

1. Reading Comprehension: Techniques for Effective Reading; Critical Reading and Analysis; Reading Technical and Non-Technical Texts

2. Vocabulary Building: Strategies for Learning New Words; Using Context Clues; Technical Vocabulary for Computer Science

3. Writing for the Web: Writing Blogs and Articles; Social Media Communication; Writing Content for Websites

TEXTBOOKS:

1. "Technical Communication" by Mike Markel
2. "English for Technical Communication" by Aysha Viswamohan
3. "Effective Technical Communication " by M Ashraf Rizvi

2nd Semester

MCPC1005 OBJECT ORIENTED PROGRAMMING USING JAVA (3-0-0)

Course Objectives:

- To provide an understanding of basic programming concepts using the Java programming language.
- To develop problem-solving skills using Java programming constructs.
- To introduce students to algorithmic thinking and program design techniques and enable students to write, compile, and debug programs in Java.

Course Outcomes (CO):

CO1: Understand the fundamental concepts of programming using the Java language.

CO2: Develop problem-solving skills through the application of programming constructs in Java and design & implement functions and algorithms to solve complex problems.

CO3: Demonstrate proficiency in using pointers, arrays, and structures in Java programming.

CO4: Apply error handling and debugging techniques to identify and resolve programming errors.

CO5: Utilize file handling mechanisms in Java for input / output operations and appreciate the importance of data structures and their implementation in Java.

Module – I

JAVA BASICS: Review of Object-oriented concepts, History of Java, Java buzz words, JVM architecture, Data types, Variables, Scope and lifetime 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

INHERITANCE AND POLYMORPHISM: Basic concepts, Types of inheritance, Member access rules, Usage of this and Super key word, Method Over loading, Method overriding, Abstract classes, Dynamic method dispatch, Usage of final key word.

PACKAGE SANDINTER FACES: Defining package, Access protection, importing packages, Defining and Implementing interfaces, and Extending interfaces. I / OSTREAMS : Concepts of streams, Stream classes-By the and Character stream, Reading console Input and Writing Console output, File Handling.

Module – III

EXCEPTION HANDLING: Exception types, Usage of Try, Catch, Throw, Throws and Finally keywords, Built-in Exceptions, Creating own Exception classes.

MULTITHREADING: Concepts of Thread, Thread lifecycle, creating threads using Thread class and Runnable interface, Synchronization, Thread priorities, Inter Thread communication.

AWTCONTROLS: The AWT class hierarchy, user interface components-Labels, Button, Text Components, Check Box, Check Box Group, Choice, List Box, Panels-Scroll Pane, Menu, Scrollbar. Working with Frame class, Colour, Font sandal out managers.

Module – IV

EVENTHANDLING: Events, Event sources, Event Listeners, Event Delegation Model (EDM), Handling Mouse and Keyboard Events, Adapter classes, Inner classes.

SWINGS: Introduction to Swings, Hierarchy of swing components. Containers, Top level containers,

JFrame, JWindow, JDialog, JPanel, JButton, JToggle Button, JCheckBox, JRadio Button, JLabel, JTextField, JTextArea, JList, JComboBox, JScrollPane.

APPLETS: Lifecycle of an Applet, Differences between Applets and Applications, Developing applets, simple applet.

Books:

1. Herberts chil dt(2010), The complete reference, 7th edition, Tata Mcgraw Hill, New Delhi
2. Programming with Java, E.Balagurusamy, McGraw-Hill Education, 6th Edition.
3. Head First Java, O' rielly publications2. T. Budd(2009), An Introduction to Object Oriented Programming, 3rd edition, Pearson Education, India.
4. J. Nino, F.A. Hosch(2002), An Introduction to programming and OOdesign using Java, John Wiley & sons, NewJersey.
5. Y.DanielLiang(2010), Introduction to Java programming, 7th edition, Pearson education, India.

MCPC 1006 SOFTWARE ENGINEERING (3-0-0)

Objectives:

- To provide the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases.
- To provide an idea of using various process models in the software industry according to given circumstances.
- To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Course Outcomes (CO):

CO1: Students will be able to decompose the given project in various phases of a lifecycle.

CO2: Students will be able to choose appropriate process model depending on the user requirements.

CO3: Students will be able perform various lifecycle activities like Analysis, Design, Implementation, Testing and Maintenance.

CO4: Students will be able to know various processes used in all the phases of the product.

CO5: Students can apply the knowledge, techniques and skills in the development of a software product.

Module – I

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software. A **Generic view of process:** Software engineering – A layered technology, aprocess framework, Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, spiral, specialized process models, The Unified process.

Module – II

Requirement analysis: Problems in information elicitation, methods of eliciting user requirements, functional and non-functional requirements, tools for requirement analysis, document flowcharts, decision tables, dataflow diagrams, data dictionaries, tools for analyzing real time systems, Use case diagrams, system sequence diagrams, CR Ccard, software requirement specification.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, the software requirements document.

Requirements engineering process: Feasibility studies, Requirement elicitation and analysis, Requirements validation, Requirements management, software equirement specification.

Module – III

Software design: The design process, Function-oriented design, Database oriented design, Object oriented design, Database design.

Coding: Code documentation, data declaration, statement construction, guidelines for input/output, efficiency with regard to code, memory and input/output.

Module – IV

Testing: Uni-testing, black box and white box testing, test cases, integration testing, top-down and bottom-uptesting, validation testing, alpha and beta testing, system testing.

Maintenance: Software reliability, availability, and maintainability, Reliability models.

Risk management: software risks, Risk identification, Risk projection, Risk refinement, Quality Management: Quality concepts, Software quality assurance, Software reliability, The ISO 9000 quality standards.

Books:

1. Software Engineering: A Practitioners Approach by Roger Pressman, 6th Edition, McGraw-Hill
2. Software Engineering by Ian Sommer ville, Addison-Wesley
3. Fundamentals of Software Engineering by Rajiv Mall, PHI

MCPC1007 DATASTRUCTURES (3-0-0)

Objectives:

- Course objectives reflects pacific knowledge, skills, abilities, or competencies that instructor suspect students to acquire from a particular course.
- Course objectives are often very specific and detailed statements that describe the content or skills that will be taught in the classroom.
- In some regards, course objective scan be thought of as inputs of student learning, representing the many important details that faculty members will cover during a particular course.

Course Outcomes (CO): After successful completion of the course the student will be able to:

CO1: To understand the role and application of Data Structure in real life.
CO2: To develop abstract data types for solving the complex problems.
CO3: To understand the concepts of non-linear data structure and application.
CO4: To analyze the efficiency of algorithms.
CO5: To describe the concept of Graph Theory in detail.

Module – I

Fundamentals: Introduction to Data Structures, Classification of Data Structures, Algorithms, Measuring Space and Time Complexities, Asymptotic Notations, Abstract Data Types.

Arrays: Storage Structures for Arrays, Sparse Matrixes, Strings, Pattern Matching.

Linked Lists: Dynamic Memory Management, Single Linked Lists, Double Linked Lists, Circular Linked Lists, Operations on Polynomials.

Stacks and Queues: Representation, Linked Stacks and Queues, Operations on Stacks and Queues, Applications of Stack and Queues.

Module – II

Trees: Terminology, Representation, Binary Trees, Binary Search Trees, Searching, Insertion and Deletions Operations in a Binary Search Tree, Height Balanced Trees, M-way Search Trees, B-Trees, B+Trees, General Trees, Representation of General Trees and Binary Trees, Forests, Application of Trees.

Module – III

Graphs: Terminology, Representation, Path Matrix, Graph Traversal, Shortest Path Problems, Topological Sort.

Searching and Sorting Techniques: Linear and Binary Search, Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge Sort, Heap and Heap Sort, Radix Sort, Compare is one of Sorting Techniques.

Module – IV

Hashing: Hash Functions and Hashing Techniques. External sorting, Implementation using programming in C.

Books:

1. Data Structures Using C-Aaron M. Tenenbaum
2. Tremblay, Jean-Paul, and Paul G. Sorenson, "An introduction to data structures with applications", McGraw-Hill, Inc.,1984.
3. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, "Fundamentals of Data Structures in C", Second Edition, 2008, Universities Press Pvt. Ltd. Hyderabad.
4. Seymour, Lipchitz. "Data Structures with C." TMH (2010).

MCPC 1008 COMPUTER ORGANIZATION AND ARCHITECTURE (3-0-0)

Objectives:

- To obtain the basic architecture land organizational concepts of a digital computer.
- To analyze performance issues in processor and memory design of a digital computer.
- To understand process or performance improvement using instruction level parallelism.

Course Outcomes (CO): After successful completion of the course the student will be able to:

CO1: Understand back ground of internal communication of computer and have better idea on how to write assembly language programs.
CO2: Be clear with memory management techniques.
CO3: Understand the communication IO devices with processor.
CO4: Notice how to perform computer arithmetic operations.
CO5: Be clear with pipeline procedure and multi processors.

Module – I

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

Module – II

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

Module – III

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.

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

Module – IV

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

Books:

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

MCPC1009 THEORY OF COMPUTATION (3-0-0)

Objectives:

1. Apply theory of computation concepts to solve problems in computer science
2. Understand the fundamental concepts of automata theory, formal languages, and computation models.
3. Analyze and design finite automata
4. Understand the basics of Theory of Computation, design and minimize finite automata
5. Study the properties of regular languages, context – free languages
6. Analyze and design pushdown automata, understand context – free grammars
7. Understand Turing machines, analyze un-decidable problems and recursively enumerable languages
8. Analyze complexity, understand formal language properties

Course Outcomes (CO):

Upon successful completion of this course, the student shall be able to:

CO1: Apply finite automata concepts to solve problems and describe the types of grammar and derivation tree.

CO2: Analyze a given Finite Automata machine and find out its Language and apply pushdown automata and context-free grammar concepts to solve problems

CO3: Apply Turing machine concepts to solve problems

CO4: Apply complexity theory and formal language property concepts to solve problems

CO5: Develop a computational model using Turing machine for the given problem. Examine the complexity for PandNP completeness for the given problem.

Module – I

Introduction to Theory of Computation, Finite Automata(FA): Deterministic FA(DFA) and Non deterministic FA(NFA), Finite Automata with Epsilon - Transition.

Module – II

Regular expressions, Finite automata and Regular expressions, Applications of regular expressions, Algebraic laws of regular expressions, Pumping Lemma and its application for regular languages, Closure and Decision properties of regular languages.

Module-III

Context - Free Grammars, Parse trees, Ambiguity in Grammar & Languages, Pushdown automation. The language of PDA. Equivalence of PDA's and CFG's. Deterministic pushdown automata, Chomsky Normal form, the pumping Lemma for context free languages, Decision properties of CFL's.

Module-IV

The Turing machine, Programming techniques for Turing machines, Extension to the basic Turing machine, Restricted Turing machine, Turing machines and computers. Non-Recursively enumerable languages, Un-decidable problem that in recursively enumerable, Un-decidable problem about Turing machines, Post's correspondence problem, other un-decidable problems.

Books:

1. Introduction to Automata Theory, Languages and Computation-J. Hoperoft, R.Motwani, J.D.Ullman – Pearson Education

Reference Books:

1. Introduction to Theory of Computation–M.Siper, Thomson Learning
2. P.Linz, "An Introduction to formal Languages and Automata", Norasa, 2000
3. Lewish Papadimitra: Theory of Computations, Prentice Hall of India, New Delhi.

MCHS1002 UNIVERSAL HUMAN VALUES & PROFESSIONAL ETHICS (2-0-0)

Objectives:

This course is intended to:

1. To assist students in recognizing the fundamental interdependence between 'VALUES' and 'SKILLS' in achieving enduring happiness and prosperity, which are the primary objectives of all individuals?
2. To promote the cultivation of a holistic perspective among students regarding life, profession, happiness, and prosperity, grounded in a accurate comprehension of human reality and the broader existence. This comprehensive view point under pins Universal Human Values and the transition towards a naturally value-oriented existence.
3. To emphasize the potential consequences of a holistic understanding regarding ethical human behavior, trustworthy and mutually satisfying interactions among individuals, and meaningful relationships with Nature.
4. This course aims to offer essential guidance in value education to in quisitive youth.

Course Outcomes (CO)

CO1 At the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature);

CO2 They would develop greater responsibility in life and in addressing issues with sustainable solutions, while considering human relationships and human nature.

CO3 They would possess enhanced critical faculties.

CO4 They would also develop sensitivity to their dedication to their understanding of human values, relationships, and society.

CO5 It is anticipated that they will apply their acquired knowledge to various real-life situations, marking a preliminary step in this direction.

Module-I

Introduction to Value Education:

Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education) Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfil the Basic Human Aspirations

Module – II

Harmony in the Human Being:

Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health

Module-III

Harmony in the Family and Society: Harmony in the Family –the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' –as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order

Module-IV

Harmony in the Nature / Existence:

Understanding Harmony in the Nature, Interconnectedness, self – regulation and Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence.

Module - V

Implications of the Holistic Understanding a Look at Professional Ethics: Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics Holistic Technologies, Production Systems and Management Models-Typical Case Studies, Strategies for Transition towards Value-based Life and Profession

Books:

- The Textbook A Foundation Course in Human Values and Professional Ethics, RR Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN978-93-87034-47-1
- The Teacher's Manual for a Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G

Reference Books:

- Jeevan Vidya: E k Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
- Human Values, A. N. Tripathi, NewAge Intl. Publishers, New Delhi, 2004.

MCPC1204 OBJECTORIENTEDPROGRAMMINGLAB (0-0-3)

List of Programs:

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 complex number 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

	000*000*
24	0*00*00*0
369	00*0*0*00
481216	000***000

8. Write a Java program to draw a pattern such as
9. Write a Java program to convert Decimal to Binary in Java
10. Writeaprogramtoaddtwotimesgiveninhourminutesandsecondsusingclassandobject.
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.

MCPC1205 SOFTWARE ENGINEERING LAB (0-0-3)

List of Experiments:

1. **Identifying the Requirements from Problem Statements.**
   Requirements, Characteristics of Requirements, Categorization of Requirements, Functional Requirements, Identifying Functional Requirements

2. **Estimation of Project Metrics**
   Project Estimation Techniques, COCOMO, Basic COCOMO Model, Intermediate COCOMO Model, Complete COCOMO Model, Advantages of COCOMO, Drawbacks of COCOMO, Halstead's Complexity Metrics.

3. **Modeling UML Use Case Diagrams and Capturing Use Case Scenarios**
   Use case diagrams, Actor, Use Case, Subject, Graphical Representation, Association between Actors and Use Cases, Use Case Relationships, Include Relationship, Extend Relationship, Generalization Relationship, Identifying Actors, Identifying Usecases, Guidelines fordrawing Use Case diagrams

4. **E-R Modeling from the Problem Statements**
   Entity Relationship Model, Entity Set and Relationship Set, Attributes of Entity, Keys, Weak Entity, Entity Generalization and Specialization, Mapping Cardinalities, E R Diagram, Graphical Notations for E R Diagram, Importance of E R modeling.

5. **Identifying Domain Classes from the Problem Statements**
   Domain Class, Traditional Techniques for Identification of Classes, Grammatical Approach Using Nouns, Advantages, Disadvantages, Using Generalization, Using Subclasses, Steps to Identify Domain Classes from Problem Statement, Advanced Concepts

6. **State chart and Activity Modeling**
State chart Diagrams, Building Blocks of a State chart Diagram, State, Transition, Action, Guidelines for drawing State chart Diagrams, Activity Diagrams, Components of an Activity Diagram, Activity, Flow, Decision, Merge, Fork, Join, Note, Partition, Guidelines for drawing an Activity Diagram

7. **Modeling UML Class Diagrams and Sequence diagrams**
Structural and Behavioral aspects, Class diagram, Elements in class diagram, Class, Relationships, Sequence diagram, Elements in sequence diagram, Object, Life-line bar, Messages

8. **Modeling Data Flow Diagrams**
Data Flow Diagram, Graphical notations for Data Flow Diagram, Explanation of Symbols used in DFD, Context diagram and leveling DFD

9. **Estimation of Test Coverage Metrics and Structural Complexity**
Control Flow Graph, Terminologies, McCabe's Cyclomatic Complexity, Computing Cyclomatic Complexity, Optimum Value of Cyclomatic Complexity, Merits, Demerits

10. **Designing Test Suites**
Software Testing, Standards for Software Test Documentation, Testing Frameworks, Need for Software Testing, Test Cases and Test Suite, Types of Software Testing, Unit Testing, Integration Testing and System Testing.

MCPC1206 DATASTRUCTURE LAB (0-0-3)

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 the following operations on Single linked list:
   i. Creation
   ii. Insertion
   iii. Deletion
   iv. Travers alin both ways
6 Implementation of the following operations on Double linked list:
   i. Creation
   ii. Insertion
   iii. Deletion
7 Implementation of Stack Using Linked List.
8 Implementation of Queue Using Linked List.
9 Implementation of the following operations on Binary Tree:
   i. Creation
   ii. Insertion
   iii. Deletion.
10 **Implementation of Binary Tree Traversal:** Preorder, In order and Post order.
11 Implementation of Binary Search Tree.
12 **Implementation of sorting algorithms:** Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort and Heap sort.

13 **Implementation of Searching Algorithms:** Linear Search and Binary Search
14 Implementation of Breadth First Search (BFS) in a Graph.
15 Implementation of Depth First Search (DFS) in a Graph.
16 Implementation of Hashing using hash functions

MCPC1207 PROGRAMMING IN PYTHON LAB (0-0-3)

List of Programs:

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 cuberoot 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 getastring made of the first 2 and last 2 characters from agivenstring. 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.

****

MCA 3rd Semester (2nd Year) New Syllabus for MCA students from 2024 onwards

FOR ADMISSION BATCH 2024-25
MASTER IN COMPUTER APPLICATION
SECOND YEAR (THIRD SEMESTER)

Sl. No.	Category	Course Code	Course	Contact Hrs. L-T-P	Credit	University Marks	Internal Evaluation
Subject (Theory)							
1.	PC	MCPC2001	Design and Analysis of Algorithms	3-0-0	3	100	50
2.	PC	MCPC2002	Operating Systems	3-0-0	3	100	50
3.	PC	MCPC2003	Artificial Intelligence and Machine Learning	3-0-0	3	100	50
4.	PE	MCPE2001	Computational Intelligence	3-0-0	3	100	50
		MCPE2002	Cryptographic Foundation				
		MCPE2003	Object Oriented Analysis and Design				
		MCPE2004	Internet & Web Technology				
		MCPE2005	Natural Language Processing				
		MCPE2006	Compiler Design				
5.	PE	MCPE2007	Soft Computing	3-0-0	3	100	50
		MCPE2008	Data Mining and Data Warehousing				
		MCPE2009	Advanced Computer Architecture				
		MCPE2010	Blockchain Technology				
		MCPE2011	Operation Research				
		MCPE2012	Software Testing and Quality Assurance				
6.	HS	HSHS2003	Entrepreneurship Development	2-0-0	2	100	50
Subject (Practical / Sessional)							
7.	PC	MCPC2201	Design and Analysis of Algorithms Laboratory	0-0-3	1.5	–	100
8.	PC	MCPC2202	Operating Systems Laboratory	0-0-3	1.5	–	100
9.	PC	MCPC2203	AI & ML Laboratory	0-0-3	1.5	–	100
10.	PSI	MCPS2201	Summer Internship and Research Experience	0-0-3	1.5	–	100
Total				**17-0-12**	**23**	**600**	**700**

Note: Click here to view/download the syllabus of the subjects.

MCPC2001 Design and Analysis of Algorithm (3-0-0)

Course Objectives:

The objective of the course is to:

- Design and analysis of algorithms and find complexity on them.
- Find the complexity of different sorting algorithm.
- Understand the dynamic programming and greedy algorithms
- Analyze and implementation of Graph Algorithms
- Demonstrate NP-Completeness and Approximation Algorithms

Module I

Introduction to design and analysis of algorithms, Growth of functions, Recurrences, Solution of recurrences by Substitution, Recursion tree and Master method, Worst case analysis of Merge sort, Quick sort and Binary search Heap sort: Heaps, Building a heap, The Heap sort algorithm, Priority Queue, Lower bounds for sorting

Module II

Dynamic Programming: Matrix-chain multiplication, Elements of dynamic programming, longest common subsequence Greedy Algorithms: An activity-selection problem, Elements of greedy strategy, Fractional knapsack problem, Huffman codes

Module III

Data structures for Disjoint Sets: Disjoint set operations, Linked-list representation of disjoint sets, Disjoint-set forests. Graph Algorithms: Elementary Graph Algorithms: Representations of graphs, Breadth-first search, Depth-first search, Minimum Spanning Trees: Kruskal and Prim's algorithms, Single-Source Shortest Paths: The Bellman-Ford and Dijkstra's algorithm, All-Pairs Shortest Paths: The Floyd-Warshall Algorithm

Module IV

Maximum Flow: Flow Networks, The Ford-Fulkerson method, Polynomials and the FFT: Representation of polynomials, The DFT and FFT, String Matching: The naïve string-matching algorithm, The Rabin-Karp algorithm.
NP-Completeness: Polynomial time, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs, NP-completeness problems, Approximation Algorithms: The vertex-cover problem, The travelling-salesman problem, The set-covering problem, The subset-sum problem.

Course Outcomes:

Upon successful completion of this course, the student shall be able to:

CO 1. Identify the different design technique of algorithm.
CO 2. Analyze the different sorting algorithm based on time and space.
CO 3. Analyze the different approaches of designing algorithm like dynamic programming and greedy algorithms
CO 4. Discuss different graph algorithm.

Books:

1. Introduction to Algorithms: T. H. Cormen, C. E. Leiserson, R. L. Rivest (PHI), Second Edition.
2. E. Horowitz, S. Sahani, S. Rajsekharan," Fundamentals of Computer Algorithms", Second Edition, Universities Press,2007

Reference Books:

1. Design and Analysis of Algorithm – M R Kabat, PHI Learning Pvt. Ltd.
2. Algorithm Design –Goodrich, Tamassia, WileyIndia.
3. Algorithms By Sanjay Dasgupta, Umesh Vazirani – McGraw-Hill Education

MCPC2002 OPERATING SYSTEMS (3-0-0)

Course Objectives

- To understand the fundamental concepts and role of Operating System.
- To learn the Process Management and Scheduling Algorithms
- To provide a detailed discussion of the various memory management techniques
- To gain insight on I/O and File management techniques
- Understand various problems related to concurrent operations and their solutions.

MODULE-1

Operating Systems: Definition, Generations of Operating systems, Types of Operating System, Functions of operating System, Abstract view of OS, System Structures, System Calls.
Processes: Definition, Process Relationship, Process states, Process State transitions, Process Control Block, Context switching, Threads, Concept of multithreads, Benefits of threads, Types of threads.

MODULE-2

Process Scheduling: Definition, Scheduling objectives, Types of Schedulers. Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time, Scheduling algorithms: Preemptive and Non-preemptive, FCFS, SJF, RR, Priority Scheduling, Multiple Queue Scheduling, Multilevel Feedback Queue Scheduling.
Deadlocks: Definition, Deadlock characteristics, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

MODULE-3

Inter-process Communication: Race Conditions, Critical Section, Mutual Exclusion, Peterson's Solution, The Producer Consumer Problem, Semaphores, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.
Memory Management: Main Memory, Swapping, Memory allocation Methods, Internal and External fragmentation and Compaction, Paging, Structure of Page Table, Segmentation, Virtual Memory: Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

MODULE-4

Disk management: Disk Structure, Disk Scheduling, RAID Structure. **I/O Management**: I/O devices, direct memory access. File Management: File concept, access methods, File types, File operation, Directory structure, Allocation methods (contiguous, linked, indexed). **Security & Protection:** Security Environment, Design Principles of Security, User Authentication.

Course Outcomes

CO 1. Understand the structure and functions of Operating System and how of the working principle of various types of operating systems
CO 2. Ability to comprehend the techniques used to implement the process manager
CO 3. Compare the performance of Scheduling Algorithms and Analyze resource management techniques
CO 4. Ability to comprehend virtual memory abstractions in operating systems

Text Book:

1. Operating System Concepts (9thEdition) by Silberschatz, Peter B. Galvin and Greg Gagne, Wiley Indian Edition.

Reference Books:

1. Modern Operating Systems (Fourth Edition) by Andrew S Tanenbaum, Prentice Hall India.
2. William Stallings, "Operating Systems Internals and Design Principles", Pearson, 2018, 9th Edition.

MCPC2003 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (3-0-0)

Course Objectives:

1. To provide a strong foundation on fundamental concepts in Artificial Intelligence
2. To Provide a basic exposition to the goals and methods of Computational Intelligence
3. To study of the design of intelligent computational techniques
4. To enable problem-solving through various machine learning techniques.

Module I

Introduction to AI - Intelligent Agents, Problem-Solving Agents, **Searching for Solutions -** Breadth-first search, Depth-first search, Hill-climbing search, simulated annealing search, Local Search in Continuous Spaces. Heuristic functions, Hill Climbing, Best First Search, A*, Adversarial Search: Game Playing, Min-Max Search, Alpha - Beta Pruning.

Module II

Knowledge and Reasoning: A Knowledge Based Agent, WUMPUS WORLD Environment, Propositional Logic, First Order Predicate Logic, Forward and Backward Chaining. Expert Systems: Introduction, Design of Expert systems.

Module III

Introduction MLP. Type of Human Learning, Type of Machine Learning: Supervised, unsupervised, reinforcement, General Model of Learning Agents

Module IV

Supervised: holdout method, K-fold cross-validation method, boot strapping, simple-regression method, unsupervised: clustering, association, reinforcement learning model.

Course Outcomes: Upon successful completion of this course, students should be able to:

CO1: Apply the Computational Intelligence techniques in applications which involve searching, reasoning and learning.

CO2: Understand artificial intelligence techniques for information retrieval

CO3: Apply Computational Intelligence techniques primarily for machine learning.

CO4: Apply the machine learning techniques for problem solving and validating

Text Books:

1. Elaine Rich, Kevin Knight, Shivshankar B Nair, Artificial Intelligence, McGraw Hill, 3rdEdition.
2. Tom Mitchell, Machine Learning, McGraw Hill , 1997,

Reference Books

1. Richard O. Duda, Peter E. Hart, David G. Stork, Pattern classification, Wiley , (2nd edition).Wiley, New York, 2001
2. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, 1st Edition,1996, PHI Learning Pvt. Ltd., New Delhi.
3. Nills J. Nilsson, “Artificial Intelligence: A New Synthesis”, 2nd Edition, 2000, Elsevier IndiaPublications, New Delhi.
4. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer , 2011 edition
5. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, MIT Press , 2016

MCPE2001 COMPUTATIONAL INTELLIGENCE (3-0-0)

Course Objectives:

5. To provide a strong foundation on fundamental concepts in Computational Intelligence
6. To Provide a basic exposition to the goals and methods of Computational Intelligence
7. To study of the design of intelligent computational techniques
8. To enable problem-solving through various searching techniques.

Module I

Introduction to Artificial Intelligence-Search-Heuristic Search-A* algorithm-Game Playing- Alpha-Beta Pruning-Expert Systems-Inference-Rules-Forward Chaining and Backward Chaining- Genetic Algorithms.

Module II

Proposition Logic — First Order Predicate Logic — Unification — Forward Chaining -Backward Chaining — Resolution — Knowledge Representation — Ontological Engineering — Categories and Objects — Events — Mental Events and Mental Objects — Reasoning Systems for Categories — Reasoning with Default Information — Prolog Programming. Non monotonic reasoning-Fuzzy Logic-Fuzzy rules-fuzzy inference-Temporal Logic-Temporal Reasoning-Neural Networks-Neuro-fuzzy Inference.

Module III

Probability basics — Bayes Rule and its Applications — Bayesian Networks — Exact and Approximate Inference in Bayesian Networks — Hidden Markov Models — Forms of Learning — Supervised Learning — Learning Decision Trees — Regression and Classification with Linear Models — Artificial Neural Networks — Nonparametric Models — Support Vector Machines — Statistical Learning — Learning with Complete Data — Learning with Hidden Variables- The EM Algorithm — Reinforcement Learning

Module IV

Natural language processing-Morphological Analysis-Syntax analysis-Semantic Analysis-AI applications — Language Models — Information Retrieval — Information Extraction — Machine Translation — Machine Learning — Symbol-Based — Machine Learning: Connectionist — Machine Learning.

Course Outcomes: Upon successful completion of this course, students should be able to:

CO1: Apply the Computational Intelligence techniques in applications which involve perception, reasoning and learning.
CO2: Apply Computational Intelligence techniques for information retrieval
CO3: Apply Computational Intelligence techniques primarily for machine learning.
CO4: Apply the Intelligent techniques for problem solving

Text Book:

1. Stuart Russell, Peter Norvig, ―Artificial Intelligence: A Modern Approach‖, Third Edition, Pearson Education / Prentice Hall of India, 2010.

Reference Books:

1. Elaine Rich and Kevin Knight, ―Artificial Intelligence‖, Third Edition, Tata McGrawHill, 2010.
2. Patrick H. Winston. "Artificial Intelligence", Third edition, Pearson Edition, 2006.
3. Dan W. Patterson, ―Introduction to Artificial Intelligence and Expert Systems‖, PHI, 2006.
4. Nils J. Nilsson, ―Artificial Intelligence: A new Synthesis‖, Harcourt Asia Pvt. Ltd., 2000.

MCPE2002 CRYPTOGRAPHIC FOUNDATIONS (3-0-0)

Course Description : The course introduces the underlying the principles and design of cryptosystems. The course covers the basics concepts of cryptography including: traditional ciphers, block ciphers, stream ciphers, public and private key cryptosystems. The course also includes the theory of hash functions, authentication systems, network security protocols and malicious logic.

Course Objectives:

After learning the course the students should be able to:

- Understand the principles and practices of cryptographic techniques.
- Understand information security goals for designing secure systems.
- Apply security algorithms in solving real-life security problems in communicating systems.
- Understand different cryptographic techniques based on asymmetric key encryption.

Module-I

Introduction: What is modern cryptography, Historical ciphers and their cryptanalysis, The heuristic versus the rigorous approach; adversarial models and principles of defining security

Perfectly-Secret Encryption: Definitions, the one-time pad; proven limitations

Private-Key (Symmetric) Encryption: Computational security, Defining secure, encryption, Constructing secure encryption; pseudo randomness, Stronger security notions, Constructing CPA-secure encryption, Modes of operation; CBC vs. CTR, Security of CTR with n − k bit counter for messages to size 2k blocks with proof directly to the LR definition, CCA attacks.

Module-II

Message Authentication Codes: Message integrity, Definition of security, Constructions from pseudorandom functions, CBC-MAC, Authenticated encryption.

Collision-Resistant Hash Functions: Definitions, The Merkle-Damgard transform, HMAC, Birthday attacks, The Random oracle model, Password hashing, Constructions of Pseudorandom Permutations (Block Ciphers) in Practice, Substitution-permutation and Feistel networks, DES and attacks on reduced-round versions, double-DES and triple-DES, AES, Hash functions from block ciphers.

Module-III

Number Theory: Preliminaries and basic group theory, Primes, factoring and RSA, Cryptographic assumptions in cyclic groups, Collision-resistant hash functions from discrete log, Public-Key (Asymmetric) Cryptography: Introduction and motivation, Diffie-Hellman key exchange

Module-IV

Public-Key (Asymmetric) Encryption: The model and definitions, Hybrid encryption and KEM/DEM, El Gamal, RSA: textbook encryption, attacks on textbook RSA, padded RSA; CCA-secure RSA KEM.Digital Signatures: Definition and applications, Hash and sign, RSA signatures: textbook RSA, hashed RSA, security with ROM, Certificates and public-key infrastructures.

Text Book

1. Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, second edition 2014, CRC Press.

Reference Books

1. Cryptography: Theory and Practice by Douglas Stinson, Third edition, CRC Press.
2. Handbook of Applied Cryptography by Alfred Menezes, Paul Oorschot and Scott Vanstone. Available Online.
3. Foundations of Cryptography by Oded Goldreich. Available Online.
4. Cryptography, an Introduction by Nigel Smart. Available Online

MCPE2003 Object Oriented Analysis & Design (3-4-4)

MCPE2004 Internet and Web Technology (3-0-0)

Course Objectives:

- To learn about the concept of Internet and a Web
- To learn how to design Website
- To learn about client scripting and server scripting language to implement client server communication

Module-1: TCP/IP Overview

TCP/IP and Internet: Layers of TCP/IP, Network Layer: Addressing, Sub netting, Introduction to WWW, WWW Architecture, URL, Domain Name, Overview of HTTP, Client server model , Web browser and Web servers , Generation of dynamic web pages, Features of Web 2.0 ,Web Hosting.

Web Design: Concepts of effective web design, Web design issues including Browser, Bandwidth and Cache, Display resolution, Look and Feel of the Web site, Page Layout and linking, User centric design, Sitemap, Planning and publishing website, Designing effective navigation

Module-2: HTML and CSS

HTML :Basics of HTML, formatting and fonts, commenting code, color , images, hyperlink, lists, tables, forms, XHTML, Meta tags, Character entities, frames and frame sets, features of HTML5. **Style sheets** : Need for CSS, introduction to CSS, basic syntax and structure, using CSS, Internal ,External and Inline style, Background images, colors and properties, Manipulatingtext ,Margins and Padding ,Positioning using CSS.

XML: Introduction to XML, uses of XML, simple XML, XML key components, DTD and Schemas

Module-3: JavaScript

JavaScript: Client side scripting with JavaScript, variables, functions, conditions, loops and repetition**Advance JavaScript**: JavaScript and objects, JavaScript Built in objects, the DOM and web browser environments, Manipulation using DOM, forms and validations

DHTML: Combining HTML, CSS and JavaScript, Events and buttons

Module-4: PHP

Introduction and basic syntax of PHP, decision and looping statement, Arrays, Functions, Browser control and detection, string handling, Form processing ,Files ,PHP with Database connectivity, Cookies and Session handling in PHP

Reference Books:

1. Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India
2. Web Technologies, Black Book, dreamtech Press
3. HTML 5, Black Book, dreamtech Press
4. The Complete Reference – PHP, Steven Holzner, Tata McGraw Hill, 2008.
5. PHP & MySQL in easy Steps, Mike Mcgrath, Tata McGraw Hill, 2012.
6. Web Design, Joel Sklar, Cengage Learning
7. Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel, Pearson

MCPE2005 NATURAL LANGUAGE PROCESSING (3-0-0)

COURSE OBJECTIVES:

1. To teach the fundamentals of NLP, and also to make them for understanding CFG, PCFG in NLP.
2. To know the role of semantics of sentences and pragmatic.
3. To teach the basic concepts of speech processing along with analysis and modeling.

MODULE-1

Introduction: Origins and challenges of NLP, Language modeling: grammar-based LM, statistical LM regular expressions, finite-state automata, English morphology, transducers for lexicon and rules, tokenization, detecting and correcting spelling errors, minimum edit distance.

Word Level Analysis: Unsmoothed n-grams, evaluating n-grams, smoothing, interpolation and backoff – word classes, part-of-speech tagging, rule-based, stochastic and transformation-based tagging, issues in pos tagging, hidden markov and maximum entropy models.

MODULE-2

Syntactic Analysis: Context free grammars, grammar rules for english, treebanks, normal forms for grammar, dependency grammar, syntactic parsing, ambiguity, dynamic programming parsing, shallow parsing, probabilistic CFG, probabilistic CYK, probabilistic lexicalized CFGs, feature structures, unification of feature structures.

MODULE-3

Semantics and Pragmatics: Requirements for representation, first-order logic, description logics, syntax-driven semantic analysis, semantic attachments, word senses, relations between senses, thematic roles, selection restrictions, Word Sense Disambiguation(WSD), WSD using supervised, dictionary & thesaurus, bootstrapping methods, word similarity using thesaurus and distributional methods.

Speech fundamentals: articulator phonetics, production and classification of speech sounds; acoustic phonetics, acoustics of speech production, review of digital signal processing concepts, short-time Fourier transform, filter- bank and LPC methods.

MODULE-4

Speech-Analysis and Speech Modeling: Features, feature extraction and pattern comparison techniques: speech distortion measures, mathematical and perceptual, log spectral distance, cepstral distances, weighted cepstral distances and filtering, likelihood distortions, spectral distortion using a warped frequency scale, lpc, plp and mfcc coefficients, time alignment and normalization, dynamic time warping, multiple time, alignment paths. Hidden markov models: markov processes, HMMS – evaluation, optimal state sequence, viterbi search, baum-welch parameter re-estimation, implementation issues.

COURSE OUTCOMES:

CO 1. Learn the fundamentals of NLP andUnderstand the use of CFG and PCFG in NLP
CO 2. Understand the role of semantics of sentences and pragmatic
CO 3. Introduce Speech Production and Related Parameters of Speech.
CO 4. Show the Computation and use of Techniques such as Short Time Fourier Transform,Linear Predictive Coefficients and other Coefficients in the Analysis of Speech.

REFERENCE BOOKS:

1. Daniel Jurafsky, James H. Martin—Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, —Natural Language Processing with Python, First Edition, OReilly Media, 2009.
3. Lawrence Rabiner And Biing-Hwang Juang, "Fundamentals Of Speech Recognition", Pearson Education, 2003.

4. Daniel Jurafsky and James H Martin, “Speech And Language Processing – An Introduction To Natural Language Processing, Computational Linguistics, And Speech Recognition”, Pearson Education, 2002

MCPE2006 COMPILER DESIGN (3-0-0)

Course Objectives: Upon successful completion of this course, students will be able to:

- Understand the Fundamental Principles and analyze Lexical Structure
- Implement Syntax Analysis, perform Semantic Analysis and design and generate suitable intermediate representations
- Apply Optimization Techniques and design Code Generators
- Utilize Compiler Construction Tools and develop a Foundational Understanding for Language Processors

Module I

Introduction, Phases of a compiler, Compiler construction tools. A simple one-pass compiler. Lexical Analysis: The role of the lexical analyzer. Input buffering. Specification of tokens: regular expressions. Recognition of tokens: finite automata (NFA and DFA). Conversion from regular expression to NFA. Conversion from NFA to DFA. Minimization of DFA. Design of a lexical analyzer generator (e.g., Lex/Flex concepts).

Module II

Introduction to Syntax Analysis, Role of the parser. Context-Free Grammars (CFG). Derivations, parse trees, ambiguity. Eliminating ambiguity. Top-Down Parsing, Recursive descent parsing. LL(1) grammars: conditions for LL(1). First and Follow sets. Construction of LL(1) parsing table. Error recovery in top-down parsing. Bottom-Up Parsing, Shift-reduce parsing. Handles and handle pruning. LR parsers: SLR, Canonical LR, LALR. Construction of LR parsing tables (conceptual overview for SLR).

Module III

Semantic Analysis: Syntax-Directed Translation (SDT): introduction to SDT schemes. Attribute grammars: synthesized attributes, inherited attributes. Dependency graphs. Type checking: type systems, static vs. dynamic checking. Symbol tables: structure, organization, operations. Run-time environment: storage organization, activation records, stack allocation. Intermediate Code Generation: Intermediate languages: three-address code (quadruples, triples, indirect triples). Postfix notation. Syntax trees.

Module IV

Code Optimization Introduction to optimization: principal sources of optimization.Basic blocks and flow graphs. Data flow analysis. Local optimizations: common subexpression elimination, dead code elimination, constant folding, strength reduction. Loop optimization: code motion, induction variable elimination. Peephole optimization. Code Generation: Issues in the design of a code generator. The target machine. Run-time storage management. Basic blocks and flow graphs (revisit). Simple code generator. Register allocation and assignment. Instruction selection.

Course Outcomes (COs):

Upon successful completion of this course, students will be able to:

CO 1: Analyze the phases of a compiler and design a lexical analyzer for a given programming language.
CO 2: Apply various parsing techniques to construct a parser for a given grammar.
CO 3: Develop syntax-directed translation schemes to perform semantic analysis and intermediate code generation.
CO 4: Apply different code optimization techniques to improve the efficiency of the generated code.

Text Book:

1. A.V. Aho, R. Sethi & J.D. Ullman "Compilers Principles Techniques and Tools" Pearson Education

Reference Books:

1. Kenneth C. Louden "Compiler Construction Principles & Practice "Cengage Learning Indian Edition

MCPE2007 Soft Computing (3-0-0)

Course Objectives:

1. To introduce the fundamental concepts of Soft Computing Techniques.
2. To understand the feasibility of applying a soft computing methodology for a particular problem.
3. To develop and understanding of Neural Networks, Fuzzy System and Genetic Algorithm.
4. To explore advanced topics such as Hybrid systems, GA based Backpropagation Networks and Fuzzy Backpropagation Networks.

MODULE - I

Introduction:

What is Soft Computing? Difference between Hard and Soft computing, Requirement of Soft computing, Major Areas of Soft Computing, Applications of Soft Computing.**Neural Networks:**What is Neural Network, Learning rules and various activation functions, Single layer Perceptrons, Back Propagation networks, Architecture of Back propagation (BP) Networks, Back propagation Learning, Variation of Standard Back propagation Neural Network, Introduction to Associative Memory, Adaptive Resonance theory and Self Organizing Map, Recent Applications.

MODULE - II

Fuzzy Systems: Fuzzy Set theory, Fuzzy versus Crisp set, Fuzzy Relation, Fuzzification, Minmax Composition, Defuzzification Method, Fuzzy Logic, Fuzzy Rule based systems, Predicate logic, Fuzzy Decision Making, Fuzzy Control Systems, Fuzzy Classification. **Genetic Algorithm:** History of Genetic Algorithms (GA), Working Principle, Various Encoding methods, Fitness function, GA Operators- Reproduction, Crossover, Mutation, Convergence of GA, Bit wise operation in GA, Multi-level Optimization.

MODULE - III

Hybrid Systems:

Sequential Hybrid Systems, Auxiliary Hybrid Systems, Embedded Hybrid Systems, Neuro-Fuzzy Hybrid Systems, Neuro-Genetic Hybrid Systems, Fuzzy-Genetic Hybrid Systems.

MODULE - IV

GA based Back propagation Networks: GA based Weight Determination, K - factor determination in Columns, **Fuzzy Backpropagation Networks:**LR type Fuzzy numbers, Fuzzy Neuron, Fuzzy BP Architecture, Learning in Fuzzy BP, Application of Fuzzy BP Networks.

Course Outcomes: Upon successful completion of this course, students should be able to:

CO1: Apply the techniques of soft computing and foster their abilities in designing and implementing soft computing-based solutions for real-world engineering problems.

CO2: Analyze neural networks to pattern recognition, classification and regression problems to evaluate solutions by various soft computing approaches.

CO3: Apply and design fuzzy logic and reasoning to handle uncertainty and solve engineering problems.

CO4: Examine and formulate genetic algorithm to combinatorial optimization problems

Text Book:

1. Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications, S.Rajasekaran, G. A. Vijayalakshami, PHI.

Reference Books:

1. Genetic Algorithms: Search and Optimization, E. Goldberg.
2. Neuro-Fuzzy Systems, Chin Teng Lin, C. S. George Lee, PHI.

MCPE2008 DATA MINING AND DATA WAREHOUSE (3-0-0)

Course Objectives:

1. To be familiar with mathematical foundations of data mining tools
2. To understand and implement classical models and algorithms in data warehouses and data mining
3. To characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
4. To master data mining techniques in various applications like social, scientific and environmental context.
5. To develop skill in selecting the appropriate data mining algorithm for solving practical problems.

MODULE I

Data Warehousing and Business Analysis: - Data warehousing Components –Building a Data warehouse –Data Warehouse Architecture – DBMS Schemas for Decision Support – Data Extraction, Cleanup, and Transformation Tools –Metadata – reporting – Query tools and Applications – Online Analytical Processing (OLAP) – OLAP and Multidimensional Data Analysis.

MODULE II

Data Mining: - Data Mining Functionalities – Data Preprocessing – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization and Concept Hierarchy Generation- Architecture of A Typical Data Mining Systems- Classification of Data Mining Systems.

Association Rule Mining: - Efficient and Scalable Frequent Item set Mining Methods – Mining Various Kinds of Association Rules – Association Mining to Correlation Analysis – Constraint-Based Association Mining.

MODULE III

Classification and Prediction: - Issues Regarding Classification and Prediction – Classification by Decision Tree Introduction – Bayesian Classification – Rule Based Classification – Classification by Back propagation – Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods – Prediction – Accuracy and Error Measures – Evaluating the Accuracy of a Classifier or Predictor – Ensemble Methods – Model Section.

MODULE IV

Cluster Analysis: - Types of Data in Cluster Analysis – A Categorization of Major Clustering Methods – Partitioning Methods – Hierarchical methods – Density-Based Methods – Grid-Based Methods – Model-Based Clustering Methods – Clustering High-Dimensional Data – Constraint-Based Cluster Analysis – Outlier Analysis.Mining Object, Spatial, Multimedia, Text and Web Data:Multidimensional Analysis and Descriptive Mining of Complex Data Objects – Spatial Data Mining – Multimedia Data Mining – Text Mining – Mining the World Wide Web.

Course Outcomes: Upon successful completion of this course, students should be able to:

CO1: Understand the functionality of the various data mining and data warehousing component
CO2: Appreciate the strengths and limitations of various data mining and data warehousing models
CO3: Perform classification and prediction of data for analyzing various data
CO4: Describe different methodologies used in data mining and data ware housing
CO5: Apply technical knowhow of the Data Mining principles and techniques for real time applications

BOOKS:

1. Jiawei Han, Micheline Kamber and Jian Pei"Data Mining Concepts and Techniques", Third Edition, Elsevier, 2011.
2. Alex Berson and Stephen J. Smith "Data Warehousing, Data Mining & OLAP", Tata McGraw – Hill Edition, Tenth Reprint 2007.
3. K.P. Soman, Shyam Diwakar and V. Ajay "Insight into Data mining Theory and Practice", Easter Economy Edition, Prentice Hall of India, 2006.
4. G. K. Gupta "Introduction to Data Mining with Case Studies", Easter Economy Edition, Prentice Hall of India, 2006.
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar "Introduction to Data Mining", Pearson Education, 2007.

MCPE2009 ADVANCED COMPUTER ARCHITECTURE (3-0-0)

Course Objectives:

This course examines the techniques and underlying principles that are used to design high-performance computers and processors. Particular emphasis is placed on understanding the trade-offs involved when making design decisions at the architectural level. A range of processor architectures are explored and contrasted. In each case we examine their merits and limitations and how ultimately the ability to scale performance is restricted.

Module-I

Principles of Processor Performance, RISC and CISC Architectures, Pipelining fundamentals, PipelineHazards, Superscalar Architecture, Super Pipelined Architecture,VLIW Architecture.

Module-II

Basic Multiprocessor Architecture: Flynn's Classification, UMA, NUMA, Distributed MemoryArchitecture, Array Processor, Vector Processors, Associative Processor, Systolic architecture.Interconnection Networks: Static Networks, Network Topologies, Dynamic Networks.

Module-III

Hierarchical Memory Technology: Data and Instruction caches, Multi-level caches, Cache memorymapping policies, Cache Coherence, Cache Performance, Virtual memory, Page replacement techniques,Memory Inter leaving, Memory Management hardware.

Module-IV

Data Flow Computer Architecture: Static Data flow computer, Dynamic Data flow computer, Clustercomputers, Distributed computing, Cloud computing.

Course Outcomes (CO):

After successful completion of the course the student will be able to:

CO1: Understand background of processor performance and different processor architectures.
CO2: Understand the multiprocessor system taxonomy.
CO3: Understand the cache performance and memory hierarchy.
CO4: Understand the advanced computer architecture systems.

Books:

1. K. Hwang and F. A. Briggs, Computer Architecture and Parallel Processing, McGraw Hill.
2. David A. Patterson and John L. Hennessy, Computer Organization and Design, Elsevier.
3. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach,Morgan Kaufmann
4. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.
5. Computer Architecture: Parhami, Oxford University Press

MCPE2010 BLOCKCHAIN TECHNOLOGY (3-0-0)

Course Objectives

This course is designed to:

- Understand how block chain systems (mainly Bit coin and Ethereum) work and to securely interact with them,
- Illustrate how to setup Ethereum tools.
- Explain the key vocabulary and concepts used in Blockchain for Business,
- Design, build, and deploy smart contracts and distributed applications.
- Integrate ideas from block chain technology into their own projects.

Module-I

Introduction to Blockchain, Concept of blockchain, History, Fundamentals of Blockchain, Characteristics of blockchain, Architecture of blockchain, transactions & chaining blocks, Public, private and Hybrid Blockchains, Distributed ledger Technologies, DLT decentralized applications and databases.

Module-II

Distributed decentralized databases, Decentralization, Hashing, Message authentication code & Secure hash Algorithms (SHA-1 & version 3), Distributed hash tables, hashing in blockchain mining, consensus approach, consensus algorithms & Byzantine agreement methods,

Module-III

Ethereum History, Ethereum virtual machine(EVM), Ethereum clients, addresses, key pairs, transaction, languages and wallets, Smart contact characteristics, Absolute and immutable, Contractual confidentiality, Cryptography primitives, Symmetric & Asymmetric cryptography.

Module-IV

Working with bitcoin, Merkle trees, Bitcoin block structure, Addresses, transactions, networks, clients, wallets and payments, Bitcoin supply, Mining Bitcoin blockchain, blocks validation and identification, blocks creation, Ming Hardware & Software, Bitcoin management and swarm. Blockchain in insurance, healthcare, blockchain in cloud computing, Artificial intelligence

Course Outcomes(CO):

After successful completion of the course student will be able to

CO1: Understand how blockchain systems (mainly Bitcoin and Ethereum)
CO2: Develop familiarity of smart contracts and decentralized application
CO3: Applications and implementation strategies
CO4: Blockchain vertical solutions, Use cases & Allied technologies

Text Books:

1. Blockchain Technology concepts and application, Wiley Publication.

Reference Books:

1. Blockchain for Enterprise Application Developers, Wiley Publication.
2. Bitcoin and Cryptocurrency Technologies, Arvind Narayanan, Princeton Publication.

MCPE2011 Operations Research (3-0-0)

Course Objectives:

1. To introduce students to the foundational concepts and techniques of operations research.
2. To develop problem-solving and analytical modelling skills.
3. To apply mathematical modelling and algorithms to optimize real-life computing and operational problems.

Module I (9 hours)

Linear Programming - Concept, Formulation, Graphical and Simplex Method, Artificial initial solution, special cases, Duality, Dual Simplex Method.

Module II: (8 hours)

Assignment Problem: Hungerian Method, **Transportation Problem**: Initial Basic solution using NWC rule, least cost and VAM, Optimal solution using MODI. **Integer Programming**-Cutting Plane, Branch & Bound.

Module-III(10 hours)

Dynamic Programming- Characteristics, Knapsack Problem, **Queuing Theory-** Basic Structure, Exponential distribution, Birth-and-Death Model, M/M/I Queue. **Game Theory**-Two-person Zero Sum game, saddle point determination, Dominance Property, Pure and mixed strategy, Solving 2x2, 2xn and mx2 games.

Module IV: (8 hours)

Sequencing- n jobs 1 machine, n-jobs 2machines, n-jobs m machines, two jobs m- machines, **Non-Linear** Programming: Unconstrained Optimization problem, Lagrange Multiplier method, Kuhn-Tucker conditions.

Course Outcomes: Upon successful completion of this course, students should be able to:

CO1: Formulate and solve linear programming problems using graphical, simplex methods and dual simplex method.
CO2: Solve the specialized LP problems like transportation and assignment using optimization techniques.
CO3: Solve IPP and apply basics of queuing theory to understand the service system.
CO4: Analyze the real life conflict situations using Game Theory and Implement decision-making and Sequencing models using OR tools in various applications.

BOOKS:

1. 1.Operations Research: An Introduction, Author: Hamdy A. Taha, Publisher: Pearson Education, Edition: 10th Edition or latest
2. Introduction to Operations Research, Authors: Frederick S. Hillier and Gerald J. LiebermanPublisher: McGraw Hill Education, Edition: Latest
3. Operations Research, Author: Kanti Swarup, P.K. Gupta, and Man Mohan, Publisher: Sultan Chand & Sons
4. Operations Research: Principles and Practice, Authors: Ravindran, Phillips, and Solberg, Publisher: Wiley India,
5. Operations Research Theory and Applications, Author: J.K. Sharma, Publisher: Macmillan

CSPE2013 SOFTWARE TESTING AND QUALITY ASSURANCE (3-0-0)

Course Objectives:

1. The objective of this course is to impart understanding of techniques for software testing and quality assurance.
2. To help students to develop skills that will enable them to construct software of high quality - software that is reliable, and that is reasonably easy to understand, modify and maintain.
3. Introduce, Understand and learn features and working of various tools of software testing and apply different software artifacts.

MODULE-1 (8 Hour)

Introduction to Software Testing: Introduction, Software Testing Process, Objectives, Software testing life cycle, Concept of testing, types of errors, Stubs and drivers, verification and validation, Definition of a Bug, Role of a Software Tester, Software Testing Axioms, Software Testing Terms and Definitions, Code inspection and code walkthrough, Testing of component based software system, Energy efficient testing.

MODULE-2 (12 Hour)

Software testing methods, Fundamentals of Software Testing: Test case design, Testing Strategies and Techniques, Structural and Functional testing, Strategic approach to software testing, Unit Testing, Integration testing , System Testing, White box testing and its types, Black box testing and its types, Static Black Box and Dynamic Black Box Testing Techniques. Special Types of Testing: Configuration Testing, Compatibility Testing, Graphical User Interface Documentation Testing, Security Testing, Test planning, Budgeting and Scheduling.

MODULE-3 (8 Hour)

Software testing Metrics, Different types of metrics, Defect Management, Defect Management Process and Metrics related to defects, Configuration and Compatibility Testing.Testing Tools: Benefits of Automation Testing, Random Testing, Bug Bashes and Beta Testing. Test Planning: Test Planning, Test Cases, Bug life cycle.

MODULE-4 (7 Hour)

Software Quality Assurance: Definition of Quality, Testing and Quality Assurance at Workplace, Test Management and Organizational Structure, Software Quality Assurance Metrics, Six Sigma, Organizational Structure: CMM (Capability Maturity Model), ISO 9000, Software Engineering Standards.

Course Outcomes:

CO1. Understand and apply knowledge of key concepts of software testing, quality and testing tools.
CO2. Design test cases and Develop test suite, write test scripts, set environmental variables for carrying out the various levels of testing manually or automatically.
CO3. Manage software defects and risks within software project.
CO4. Understand software quality and software Engineering Standards

Reference Books:

1. Kshirsagar Naik and Priyadarshi Tripathy, Software Testing & Quality Assurance- Theory and Practice, Wiley Student edition
2. William E. Perry, Effective Methods for Software Testing, WILLEY, 3rd Edition
3. Alan C. Gillies, "Software Quality: Theory and Management", International Thomson Computer Press, 1997.
4. M G Limaye, Software Testing, Tata McGraw-Hill Education, 2009