SYLLABUS

FOR

FIRST YEAR

MCA PROGRAMME

(Effective from 2024-25)

2024-25

BIJU PATNAIK UNIVERSITY OF TECHNOLOGY, ODISHA

ROURKELA

COURSE STRUCTURE

FIRST SEMESTER

Category	Subject Code	Subject	L-T-P	Credit	University Marks	Internal Marks
THEORY						
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
Total			**17-0-0**	**17**	**600**	**300**
SESSIONAL / PRACTICAL						
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**	**06**	**-**	**400**
		Total Semester		**23**	**600**	**700**
		Grand Total (Theory + Practical) = 1300				

SECOND SEMESTER

Category	Subject Code	Subject	L-T-P	Credit	University Marks	Internal Marks
THEORY						
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
Total			**17-0-0**	**17**	**600**	**300**
SESSIONAL / PRACTICAL						
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**	**06**	**-**	**400**
		Total Semester		**23**	**600**	**700**
		Grand Total (Theory + Practical) = 1300				

MCBS1001 DISCRETE MATHEMATICS (3-0-0)

Course Objectives:

1. To learn the mathematical foundations required for computer science.
2. 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

Module 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.

Module 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.

Module 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.

Module 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.

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.

Module 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

Module 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

Module 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

Module 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

MCHS1201 LANGUAGE LAB (0-0-3)

These lab experiments aim to provide practical, hands-on experience in various aspects of communicative English, tailored to the needs of MCA students.

List of Experiments:

Module 1: BASICS OF COMMUNICATION

Experiment 1: Communication Role Play
Objective: Understand verbal and non-verbal communication.
Activities:

1. Role-play different scenarios (e.g., a business meeting, a social gathering) focusing on body language, gestures, and spoken words.
2. Identify and discuss the barriers encountered.

Experiment 2: Active Listening Exercises
Objective: Enhance listening skills.
Activities:

1. Listen to a recorded speech and answer comprehension questions.
2. Engage in a paired listening activity where one student speaks and the other practices active listening, followed by feedback.

Module 2: PROFESSIONAL COMMUNICATION

Experiment 3: Email Writing Practice
Objective: Develop proficiency in writing professional emails.
Activities:

1. Write emails for different purposes (e.g., inquiry, complaint).
2. Peer review and discuss the format, tone, and etiquette of each email.

Experiment 4: Presentation Preparation and Delivery
Objective: Improve presentation skills.
Activities:
1. Prepare a PowerPoint presentation on a technical topic.
2. Deliver the presentation to the class, focusing on clarity, confidence, and use of visual aids. Receive and give constructive feedback.

Module 3: INTERPERSONAL SKILLS
Experiment 5: Team Communication Simulation
Objective: Enhance teamwork and collaboration skills.
Activities:
1. Engage in a group project simulation where roles and responsibilities are assigned.
2. Use collaborative tools (e.g., Google Docs) to work together and present findings.

Experiment 6: Mock Interviews
Objective: Prepare for job interviews.
Activities:
1. Participate in mock interviews with common interview questions.
2. Receive feedback on responses, body language, and overall performance from peers and instructor.

Module 4: ENHANCING LANGUAGE SKILLS
Experiment 7: Critical Reading Analysis
Objective: Improve critical reading skills.
Activities:
1. Read a technical article and identify key points, arguments, and conclusions.
2. Discuss the article in groups, focusing on analysis and interpretation.

Experiment 8: Vocabulary Building Exercises
Objective: Expand vocabulary.
Activities:
1. Use context clues to understand and define new technical terms from computer science texts.
2. Create flashcards for new vocabulary and engage in peer quizzes.

Experiment 9: Writing a Technical Blog
Objective: Develop web writing skills.
Activities:
1. Write a blog post on a recent technological advancement or trend.
2. Peers review the posts focusing on clarity, conciseness, and engagement.

Experiment 10: Social Media Communication
Objective: Practice concise and effective writing for social media.
Activities:
1. Create social media posts (e.g., tweets, LinkedIn updates) about a technical topic.
2. Discuss the effectiveness and engagement of each post, considering the target audience.

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.

Module 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.

Module 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).

Module 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)

Module 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:

1. To provide an understanding of basic programming concepts using the C programming language.
2. To develop problem-solving skills using C programming constructs.
3. To introduce students to algorithmic thinking and program design techniques.
4. 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.

Module 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.

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

Module 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.

Module 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.

Module– 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.

Module– 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.

Module- 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.

Module– 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

MCPC1201 COMPUTER NETWORKS LABORATORY (0-0-3)

Objective:
The Computer Networks Laboratory course aims to provide hands-on experience with the principles and practice of computer networks, focusing on both the theoretical and practical aspects of network design, implementation, and troubleshooting.

Course Outcomes:
By the end of the course, students will be able to:
1. Understand and implement various networking protocols.
2. Configure and troubleshoot network devices.
3. Analyze network performance.
4. Design and implement small-scale networks.

Laboratory Sessions:
1. Introduction to Network Lab Tools:
   - Overview of network simulation tools like Cisco Packet Tracer, GNS3, Wireshark.
   - Introduction to basic networking commands (ping, tracert, ipconfig/ ifconfig).
2. Basic Network Configuration:
   - Setting up a simple peer-to-peer network.
   - Configuring IP addresses and subnet masks.
   - Testing network connectivity using ping and tracert.
3. Error Detection and Correction:
   - Implementing CRC error detection.
   - Simulating error correction mechanisms.
4. Elementary Data Link Protocols:
   - Simulation of sliding window protocols.
   - Analysis of protocol performance over noisy and noiseless channels.
5. Medium Access Control:
   - Configuring and analyzing Ethernet networks.
   - Setting up and testing Wireless LAN (WLAN) connections.
   - Exploring Bluetooth network configurations.
6. Network Devices Configuration:
   - Setting up and configuring switches, routers, and gateways.
   - Understanding the use of repeaters, hubs, and bridges in a network.
7. Multiplexing Techniques:
   - Implementing and analyzing different types of multiplexing (TDM, FDM).
8. Routing Algorithms:
   - Implementing and analyzing shortest path routing algorithms.
   - Configuring Distance Vector Routing (RIP) and Link State Routing (OSPF).
9. IP Addressing and Subnetting:
   - Configuring IPv4 and IPv6 addressing.
   - Subnetting practice and exercises.
10. Address Mapping Protocols:
    - Implementing and analyzing ARP, RARP, ICMP, IGMP.
    - Configuring and testing DHCP.

11. Transport Layer Protocols:
    - o Simulation and analysis of TCP and UDP.
    - o Configuring TCP sliding window and congestion control mechanisms.
12. Quality of Service (QoS):
    - o Implementing and analyzing QoS in networks.
    - o Configuring QoS settings on network devices.
13. Application Layer Protocols:
    - o Setting up and testing HTTP, FTP, and DNS.
    - o Configuring and analyzing email protocols (SMTP, POP3, IMAP).
    - o Exploring TELNET and SSH for remote connectivity.

Reference Material:

- "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross.
- "Data Communications and Networking" by Behrouz A. Forouzan.
- Cisco Packet Tracer and GNS3 Documentation.

MCPC1202 C PROGRAMMING LAB (0-0-3)

List of Experiments:

1. Simple C programs.
2. Using If and switch constructs programs
3. Looping related problems
4. Programs using functions
5. If statement, If..else statement, nesting if else statement, else if ladder, switch statement, goto statement, while
6. statement, do statement, for statement
7. One-dimensional arrays, two dimensional arrays, multi dimensional arrays
8. Initialization of string variables, reading and writing strings, string handling functions
9. Programs using structures
10. Programs using unions
11. Initialization of pointer variables, address of variable, accessing a variable through its pointer
12. Pointer as Functions
13. Strings with Pointer: pointers and character strings, pointers and structures
14. Programs based on file handling
15. Command Line Arguments
16. Error Handling

MCPC1203 DATABASE MANAGEMENT SYSTEMS LAB (0-0-3)

List of Experiments:

1. Execute a single line and group functions for a table.
2. Execute DCL and TCL Commands.
3. Implement the query in SQL for a) insertion b) retrieval c) updating d) deletion.
4. Using Joins, Index, Key constraints and Normalization
5. Create views, partitions and locks for a particular DB
6. Write PL/SQL procedure for an application using exception handling
7. Write PL/SQL procedure for an application using cursors.

8. Write a DBMS program to prepare reports for an application using functions.
9. Write a PL/SQL block for transaction operations of a typical application using triggers.
10. Write a PL/SQL block for transaction operations of a typical application using package.
11. Design and develop an application using any front end and back end tool (make use of ER diagram and DFD).
12. Writing Assertion
13. Implementing operation on relation using PL/SQL
14. Creating Forms
15. Generating Reports

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 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

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
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. 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.

Module-IV
EVENT HANDLING: 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, JToggleButton, JCheckBox, JRadioButton, JLabel,JTextField, JTextArea, JList, JComboBox, JScrollPane. APPLETS: Life cycle of an Applet, Differences between Applets and Applications, Developing applets, simple applet.

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.
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.

MCPC1006 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 life cycle 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, a process 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 flow charts, decision tables, data flow diagrams, data dictionaries, tools for analyzing real time systems, Use case diagrams, system sequence diagrams, CRC card, software requirement specification.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, the software requirements document. Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management, software requirement specification.

Module-III
Software design: The design process, Function-oriented design, Data base oriented design, Object oriented design, Data base design. Coding: Code documentation, data declaration, statement construction, guidelines for input/output, efficiency with regard to code, memory and input/output.

Module-IV
Testing: Unit testing, black box and white box testing, test cases, integration testing, top-down and bottom-up testing, 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 Sommerville, Addison-Wesley
3. Fundamentals of Software Engineering by Rajiv Mall, PHI

MCPC1007 DATA STRUCTURES (3-0-0)

Objectives:
- Course objectives reflect specific knowledge, skills, abilities, or competencies that instructors expect 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 objectives can 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, Comparison 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, SartajSahni, 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).

MCPC1008 COMPUTER ORGANIZATION AND ARCHITECTURE (3-0-0)

Objectives:

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

Course Outcomes(CO):

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

CO1: Understand background 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. Pipeline 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, Superpipelined and VLIW Processor architectures. Array and Vector processors.

Module-IV

Multiprocessor 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 Vranesic and Safwat Zaky, "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 undecidable 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 P and NP completeness for the given problem.

Module-I

Introduction to Theory of Computation, Finite Automata (FA): Deterministic FA (DFA) and Nondeterministic 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, Undecidable problem that in recursively enumerable, Undecidable problem about Turing machines, Post's correspondence problem, other undecidable 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 an accurate comprehension of human reality and the broader existence. This comprehensive viewpoint underpins 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 inquisitive 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, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978- 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: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amar kantak, 1999.
- Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.

MCPC1204 OBJECT ORIENTED PROGRAMMING LAB (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
8. Write a Java program to draw a pattern such as

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

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 Use cases, Guidelines for drawing 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, ER Diagram, Graphical Notations for ER Diagram, Importance of ER 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. Statechart and Activity Modeling
   Statechart Diagrams, Building Blocks of a Statechart Diagram, State, Transition, Action, Guidelines for drawing Statechart 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 DATA STRUCTURE 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) Traversal in 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, Inorder and Postorder.
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 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.