हिमाचल प्रदेश केन्द्रीय विश्वविद्यालय
Central University of Himachal Pradesh
(Established under Central Universities Act 2009)
Shahpur Parishar, District Kangra, Himachal Pradesh-176206

Course Contents and Course Learning Outcomes of MCA (2 year)

Course Outcomes of MCA (2 year)

MCA 501
Data Structures

Course Objectives:
Using computer science theory, students will construct and analyze various data structuresand abstract data types including lists, stacks, queues, trees, and graphs. Students willimplement various sorting, searching, and hashing algorithms. Students will build asubstantial, complex data structure.

Course Level Learning Outcomes:
Upon successful completion of the course students will be able to:

- Design correct programs to solve problems.
- Choose efficient data structures and apply them to solve problems.
- Analyze the efficiency of programs based on time complexity.
- Prove the correctness of a program using loop invariants, preconditions and post conditions in programs

UNIT- I
Introduction: Basic Terminology, Data structures and its classification, Algorithm, Complexityspace & time complexity, complexity notations- big Oh, Omega, Theta. Array Definition,Representation and Analysis of Arrays, Single and Multidimensional Arrays, Address calculation,Linear Search, Binary Search of Array, Traversing, Insertion & deletion in array, Sparse Matrices,Strings. Internal and External sorting, Insertion Sort, Bubble Sort, selection sort, Quick Sort, MergeSort, Radix sort.

UNIT- II
Linked List Introduction, Representation of linked list in to memory, Memory allocation – GarbageCollection, Traversing & Searching in Linked List, Insertion into linked list- at beginning of list &at given location, Deletion in linked list- from starting of list & given location of node, HeaderLinked List, two way List, Input & output restricted liked list, Circular Header Linked List,Representation of Polynomials using linked List.

UNIT- III
Stack, Array Implementation of stack, Linked Representation of Stack, Application of stack: Conversion of Infix to Prefix and Postfix Expressions and Expression evaluation. Queue, Array and linked implementation of queues, Circular queues, D-queues and Priority Queues.

UNIT- IV
Trees: Basic terminology, Binary Trees, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Binary Search Tree (BST), AVL Trees, B-trees. Graphs: Introduction, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning

Trees, Minimum Cost Spanning Trees. Searching & Hashing: Sequential search, binary search, Hash Table, Hash Functions, Collision Resolution Strategies.

Text Books:

1. Lipschuitz L. Seymour, "Data Structures", Schaum Outline Series, TMH.
2. R. S. Salaria, "Data Structures & Algorithm Using C", Khanna Book Publishing.

Reference Books:

1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd., NDelhi.
2. R. S. Salaria, "Data Structures & Algorithm Using C++", Khanna Book Publishing
3. A.M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd., NewDelhi.
4. Trembley and Sorenson, "Data Structures", TMH Publications

MCA 502
Data Structures Lab

Course Objectives:

An objective of this course to understand the concept of data structures through ADT including List, Stack, Queues . The course will cover design and implement various data structure algorithms.

Course Level Learning Outcomes:

Upon successful completion of the course students will be able to:

- Select appropriate data structures as applied to specified problem definition.
- Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
- Students will be able to implement Linear and Non-Linear data structures.
- Implement appropriate sorting/searching technique for given problem.
- Design advance data structure using Non-Linear data structure.
- Determine and analyze the complexity of given Algorithms.

Course Contents: As per theory course of data structure.

Text Books:

- Lipschuitz L. Seymour, "Data Structures", Schaum Outline Series, TMH.

2. R. S. Salaria, "Data Structures & Algorithm Using C", Khanna Book Publishing.

Reference Books:

1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd., NDelhi.
2. R. S. Salaria, "Data Structures & Algorithm Using C++", Khanna Book Publishing
3. A.M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd., NewDelhi.
4. Trembley and Sorenson, "Data Structures", TMH Publications

MCA 503
Operating System

Course Objectives:

Aobjective of this course to understand the basic concepts of operating system. The course willcover an introduction on the policies for scheduling, deadlocks, memory management,synchronization, system calls, I/O management, and file systems.

Course Level Learning Outcomes:
Upon successful completion of the course students will be able to:

- Demonstrate the important library functions and system calls of various operating systems.
- Describe the inner workings of various operating systems.
- Write shell scripts to perform repetitive tasks.
- Design and implement shell functions.

UNIT-I
Operating System Introduction: function, characteristics, structures–simple batch,multi programmed, timeshared, personal computer, parallel, distributed systems, real-timesystems, system components, operating system services, system calls, virtual machines.

Process and CPU Scheduling: Process concepts and scheduling, operation on processes,cooperating processes, threads and inter-process communication scheduling criteria,scheduling algorithm, multiple-processor scheduling, real time scheduling.

UNIT-II
Management and Virtual memory:logical versus physical address space, swapping, contiguous allocation, paging, segmentation, segmentation with paging. Demand paging, performance of denuding paging, page replacement, page replacement algorithm, allocation of frames, thrashing.

UNIT-III
File System Interface and Implementation:access methods, directory, structure, protection, file system structure, allocation methods, free space management, directory management, directory implementation, efficiency and performance.
I/O Management: I/O software and its types, disk scheduling.
Process Management and Synchronization: Critical section problem, synchronization, critical regions, monitors.

UNIT-IV
Deadlocks:system model, dead locks characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection and recovery from deadlock. **Shell Programming:**vi editor, shell variables, I/O in shell, control structures, loops, subprograms, creating shell scripts. Basic system administration in Linux/Unix.

Text Books:

1. Silberschart, Galvin, Gagne, “Operating System Concepts”, Ninth Edition, WSE Wiley.
2. Das, S., Your UNIX: The Ultimate Guide, Fourth Edition, McGraw-Hill Inc.

Reference Book:

1. D.M. Dhamdhere, “Operating Systems: A Concept Based Approach”, Tata McGraw-Hill.
2. Milan Milenkovic, “Operating system-concepts and design”, McGraw Hill International Edition
3. A. S. Godbole, “Operating systems”, Tata McGraw hill
4.Deitel H. M., “Operating System”, Pearson Publications
5. William Stallings, “Operating Systems: Internals and Design Principles”, Prentice-Hall of India
7. Andrew. S. Tanenbaum, “Modern operating systems”, Pearson Prentice Hall

MCA-521
Software Engineering

Course Objective: The objective of this course is to provide a solid fundamental knowledge of software engineering. On completion of the course, the student is expected to work as an individual and/or in team to develop and deliver quality software.

Course Outcomes:
After completing the course, the student should be able to:

- Demonstrate knowledge of software engineering's layered technology and software process models, which serve as the foundation for the software development lifecycle
- Define the procedures involved in the discovery and documentation of software/system requirements and grasp the underlying mechanisms in the discovery and production of these requirements
- Verification and validation, static analysis, reviews, inspections, and audits are all used to test software
- Project planning, project estimation methodologies, risk management, quality management, and configuration management are all aspects of software project management

UNIT-I
Introduction: Introduction to software Engineering, Software characteristics, Software components, Software applications, Software Engineering Principles, Software metrics and measurement, monitoring and control. Software development life-cycle, Water fall model, prototyping model, Incremental model, Iterative enhancement Model, Spiral model.

UNIT-II
Software Requirement Specification: Requirements Elicitation Techniques, Requirements analysis, Models for Requirements analysis, requirements specification, requirements validation.
System Design: Design Principles: Problem partitioning, abstraction, Top down and bottom up design, structured approach. Functional versus object-oriented approach of design, design specification, Cohesiveness and Coupling.Overview of SA/SD Methodology, structured analysis, data flow diagrams, extending DFD to structure chart.

UNIT-III
Software project Management: Project planning and Project scheduling. Software Metrics: Size

Metrics like LOC, Token Count, and Function Count. Cost estimation using models like COCOMO. Risk management activities. Software Reliability and Quality Assurance: Reliability issues, Reliability metrics, reliability models, Software quality, ISO 9000 certification for software industry, SEI capability maturity model.

UNIT-IV

Testing: Verification and validation, code inspection, test plan, test case specification. Level of testing: Unit, Integration Testing, Top down and bottom up integration testing, Alpha and Beta testing, System testing and debugging. Functional testing, structural testing, Software testing strategies. Software Maintenance: Structured Vs unstructured maintenance, Maintenance Models, Configuration Management, Reverse Engineering, Software Re-engineering.

Text Books:

1. K. K. Aggarwal& Second Edition, New Age International Publishers.
2. Pankajjalote "software engineering " Wiley India

Reference Books:

1. Roger S. Pressman, " Software engineering- A Practitioner Approach", TMH
2. Rajib Mall, "Fundamental of Software Engineering", PHI Learning Pvt. Ltd.

MCA534
Object Oriented Programming

Course Objectives:

The objective of this course is to develop programming skills of students, using objectoriented programming concepts, learn the concept of class and object using C++ anddevelop simple applications using these concepts.

Course Level Learning Outcomes:

Upon successful completion of the course students will be able to:

- Analyze fundamentals of programming such as variables, conditional and iterativeexecution, methods, etc.
- Discuss the fundamentals of object-oriented programming in Java, including definingclasses, invoking methods, using class libraries, etc.
- Summarize important topics and principles of software development.
- Practice on writing a computer program to solve specific problems.

UNIT-I:

Data Types, Identifiers, Variables Constants and Literals, Basic input/output statements, Operators, Expressions, Type conversion, Control structures, Arrays, Strings, Structures and Pointers.

Functions: Basic, Recursive functions, Overloaded functions, inline functions, function with default arguments.

UNIT-II:

Introduction to classes and objects, Access specifiers, Constructor, Destructor, Function overloading, Operator overloading, friend functions.

UNIT-III:

Inheritance-Concept of derived and base class, accessing base class members, Single inheritance, multiple inheritance, hierarchical inheritance, multilevel inheritance, hybrid inheritance, constructor in derived classes.

Virtual Functions-Functions accessed with pointers, virtual member functions accessing with pointers, late binding, pure virtual functions, abstract classes, virtual base classes.

UNIT-IV:

Exception handling.

Working with files- classes for file stream operations, opening and closing a file, detecting end-offile, file modes, file pointers and their manipulations, sequential input and output operations, updating a file, error handling.

Command line arguments.

Text Book:

1. Balagurusamy, E. “Object Oriented Programming with C++”, 8E , Tata McGraw Hill.

Reference Book:

1. Herbert Schildt, “ C++ The Complete Reference ” , 4E , TMH Publication.
2. RobertLafore, “Object Oriented Programming in Turbo C++”, 4E ,Galgotia Publications Pvt. Ltd.

MCA-C09
Data Base Management System

Course Objectives:

This course is intended to provide an introduction to the management of database systems.The course emphasizes the understanding of the fundamentals of relational systemsincluding data models, database architectures, and database manipulations. The course usesa problem-based approach to learning.

Course Level Learning Outcomes:

Upon successful completion of the course students will be able to:

- Elaborate on different issues involved in the design and implementation of a databasesystem. Study the physical and logical database designs, database modeling, relational,hierarchical, and network models.
- Analysis and Practice on data manipulation language to query, update and manage adatabase.
- Determine essential DBMS concepts such as database security, integrity, concurrency,distributed database, and intelligent database, Client/Server (Database Server), DataWarehousing.
- Develop a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

UNIT-I:

Basic Concepts: Entity, Relationship and its types, Components of a database, three level architecture ofa DBMS, Database models.File Organization: Serial, Sequential, Index Sequential and Direct file organization.

UNIT-II:

Entity-Relationship Model: Entity Types, Entity Sets, Attributes & keys, Relationships,Relationships Types, Roles and Structural Constraints, Design issues, E-R Diagrams,Design of an E-R Database Schema, Reduction of an E-R Schema to Tables.

Relational Data Model: Relational model concepts, Integrity constraints over Relations,Relational Algebra – Basic Operations

UNIT-III:

SQL:DDL, DML, and DCL, views & Queries in SQL, Specifying Constraints & Indexesin SQL.

Relational Database Design: Functional Dependencies, Normalization.

UNIT-IV:

Transaction Processing Concepts: Introduction to Transaction Processing, Transaction &System Concepts, Properties of Transaction, Schedules and Recoverability, Serializabilityof Schedules.

Concurrency Control Techniques: Locking Techniques, Timestamp ordering, Multiversion Techniques, Optimistic Techniques, Granularity of Data items.

Protection & Security.

Text Books:

1. R. Elmasri and S. B. Navathe, ―Fundamentals of Database Systems‖, Seventh Edition, AddisonWesley.
2. Ivan Bayross, ―SQL, PL/SQL: The Programming Language of Oracle‖, Fourth Edition, BPB Publications.

Reference Books:

1. R. Ramakrishnan and J. Gehrke, ―Database Management Systems‖, Third Edition Tata McGrawHill.
2. A. Silberschatz, H. Korth and S. Sudarshan, ―Database System Concepts‖,Sixth Edition, Tata McGraw Hill.

MCA-C10
DBMS Lab

Course Objectives:

The course emphasizes the understanding of the fundamentals of SQL and PL/SQL.

Course Level Learning Outcomes:

Upon successful completion of the course students will be able to:

- Analysis and Practice on data manipulation language to query update and manage a database.
- Develop a complete database system for real world. It also involves modeling, designing, and implementing a Database.

UNIT-I

SQL: Data Types, Operators, DDL, DML, DCL, and TCL Commands.

UNIT-II

SQL: Integrity Constraints, Functions, Join, Indexes, Subqueries, Views.

UNIT-III

PL/SQL: Variables, Constants, Control Statements, Procedure, Functions.

UNIT-IV

PL/SQL: Cursor, Triggers, Exception Handling, Packages.

Text Books:

1. Ivan Bayross, ―SQL, PL/SQL: The Programming Language of Oracle‖, Fourth Edition, BPB Publications

MCA-C12
Analysis & Design of Algorithms

Course Objectives:

The objective of this course is to reinforce basic design concepts (e.g., pseudo code, specifications, top-down design) and have the knowledge of algorithm design strategies.This course emphasizes mainly on the analysis of an algorithm w.r.t. time and spacecomplexity.

Course Level Learning Outcomes:

Upon successful completion of the course students will be able to:

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate familiarity with major algorithms and data structures.
- Relate the real-life problems and their better solution

UNIT- I

Algorithms, designing algorithms, analyzing algorithms, asymptotic notations, Introduction to divide andconquer technique, analysis, design and comparison of various algorithms based on this technique, example binary search, merge sort, quick sort, strassen‘s matrix multiplication.

UNIT- II

Study of Greedy strategy, examples of greedy method like optimal merge patterns, Huffman coding, minimum spanning trees, knapsack problem, job sequencing with deadlines, single source shortest pathalgorithm.

Concept of dynamic programming, problems based on this approach such as 0/1 knapsack, multistage graph, reliability design, Floyd-Warshall algorithm.

UNIT- III

Backtracking concept and its examples like 8 queen‘s problem, Hamiltonian cycle, Graph coloring problem etc. Introduction to branch & bound method, examples of branch and bound method like travelling salesman problem etc.Meaning of lower bound theory and its use in solving algebraic problem.

UNIT- IV

Binary search trees, height balanced trees, 2-3 trees, B-trees, basic search and traversal techniques for trees and graphs (In order, preorder, postorder, DFS, BFS), NP-completeness.

Text Books:

1. Ellis Horowitz, SartajSahni and SanguthevarRajasekaran, ―Computer Algorithms‖, Second Edition, Universities Press.
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., ―Introduction to Algorithms‖, SecondEdition, Prentice Hall of India Pvt. Ltd.

Reference Books:

1. A.V. Aho, J.E. Hopcroft and J.D. Ullman, ―The Design and Analysis of Computer Algorithms‖,Pearson Education.
2. Sara Baase and Allen Van Gelder, ―Computer Algorithms, Introduction to Design and Analysis,Third Edition, Pearson Education.
3. S. Dasgupta, C. Papadimitriou & U. Vazirani, ―Algorithms‖; Tata McGraw Hill.
4. Michael T Goodrich, RobartoTamassia, ―Algorithm Design: Foundations, Analysis and InternetExamples, Wiley India.

MCA-OC3
Software Engineering

Course Objective: The objective of this course is to provide a solid fundamental knowledge of software engineering. On completion of the course, the student is expected to work as an individual and/or in team to develop and deliver quality software.

Course Outcomes:

After completing the course, the student should be able to:

- Demonstrate knowledge of software engineering's layered technology and software process models, which serve as the foundation for the software development lifecycle
- Define the procedures involved in the discovery and documentation of software/system requirements and grasp the underlying mechanisms in the discovery and production of these requirements
- Verification and validation, static analysis, reviews, inspections, and audits are all used to test software
- Project planning, project estimation methodologies, risk management, quality management, and configuration management are all aspects of software project management

UNIT-I

Introduction: Introduction to software Engineering, Software characteristics, Software components, Software applications, Software Engineering Principles, Software metrics and measurement, monitoring and control. Software development life-cycle, Water fall model, prototyping model, Incremental model, Iterative enhancement Model, Spiral model.

UNIT-II

Software Requirement Specification: Requirements Elicitation Techniques, Requirements analysis, Models for Requirements analysis, requirements specification, requirements validation.

System Design: Design Principles: Problem partitioning, abstraction, Top down and bottom up design, structured approach. Functional versus object-oriented approach of design, design specification, Cohesiveness and Coupling.Overview of SA/SD Methodology, structured analysis, data flow diagrams, extending DFD to structure chart.

UNIT-III

Software project Management: Project planning and Project scheduling. Software Metrics: Size Metrics like LOC, Token Count, and Function Count. Cost estimation using models like COCOMO. Risk management activities. Software Reliability and Quality Assurance: Reliability issues, Reliability metrics, reliability models, Software quality, ISO 9000 certification for software industry, SEI capability maturity model.

UNIT-IV

Testing: Verification and validation, code inspection, test plan, test case specification. Level of testing: Unit, Integration Testing, Top down and bottom up integration testing, Alpha and Beta testing, System testing and debugging. Functional testing, structural testing, Software testing strategies. Software Maintenance: Structured Vs unstructured maintenance, Maintenance Models, Configuration Management, Reverse Engineering, Software Re-engineering.

Text Books:

3. K. K. Aggarwal& Second Edition, New Age International Publishers.
4. Pankajjalote "software engineering " Wiley India

Reference Books:

3. Roger S. Pressman, " Software engineering- A Practitioner Approach", TMH
4. Rajib Mall, "Fundamental of Software Engineering", PHI Learning Pvt. Ltd.

MCA-E07
Internet of Things

Course Objective:

The Internet of things (IoT) refers to uniquely identifiable objects or things in an internet-like infrastructure, as well as their virtual representations. The Internet of Things enables things to be sensed and/or controlled remotely over existing network infrastructure, allowing for more direct integration of the physical world into computer-based systems and, as a result, increased efficiency, accuracy, and financial boost.

Course Outcomes

After completing the course, the student should be able to:

- Understand the application area of IoT
- Realize the revolutions of internet in mobile devices, cloud and sensor network
- Understand building blocks of internet of things and characteristics
- Enable the interconnection and integration of the physical world and the cyber space

UNIT-I

Introduction to IoT: Sensing, Actuation, Networking basics, Communication Protocols, Sensor Networks, Machine-to-Machine Communications, IoT Definition, Characteristics. IoT Functional Blocks, Physical design of IoT, Logical design of IoT, Communication models & APIs.

UNIT-II

M2M to IoT-The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics. Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT. M2M vsIoT An Architectural Overview–Building architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. Reference Architecture and Reference Model of IoT

UNIT-III

IoT Reference Architecture- Getting Familiar with IoT Architecture, Various architectural views of IoT such as Functional, Information, Operational and Deployment. Constraints affecting design in IoT world- Introduction, Technical design Constraints. Domain specific applications of IoT: Home automation, Industry applications, Surveillance applications, Other IoT application.

UNIT-IV

Developing IoT solutions: Introduction to Python, Introduction to different IoT tools, Introduction to Arduino and Raspberry Pi, Data Aggregation for the IoT in Smart Cities, Privacy and Security Issues in IoT.

Text Books:

1. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-onApproach)”, 1E, VPT
2. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, StamatisKarnouskos,David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a NewAge of Intelligence”, 1st Edition, Academic Press

Reference Books:

1. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to ConnectingEverything”, 1st Edition, Apress Publications
2. CunoPfister, Getting Started with the Internet of Things, O‟Reilly

MCA-SD4
Introduction to Latex

Course Objectives:

The course is designed to provide knowledge of Latex. Students will be able to use various feature of latex to write articles and research papers.

Course Level Learning Outcomes:

Upon successful completion of the course students will be able to:

- Write the article with different features like equations, tables.
- Use the different packages available in Latex.
- Write resume, Question papers and articles.

UNIT-I

Installation of the software LaTeX, Understanding Latex compilation Basic Syntax, Writing equations, Matrix, Tables.

UNIT-II

Page Layout – Titles, Abstract Chapters, Sections, References, Equation references, citation.

List making environments Table of contents, generating new commands, Figure handling numbering, List of figures, List of tables, Generating index.

UNIT-III

Packages: Geometry, Hyperref, amsmath, amssymb, algorithms, algorithmic graphic, color, tiles listing.

UNIT-IV

Classes: article, book, report, beamer, slides. IEEEtran.
Applications to: Writing Resume, Writing question paper, Writing articles/ research papers.

Text Book:

1. Leslie Lamport, ―LaTeX: A Document Preparation System, Second Edition,Addison Wesley.

Reference Book:

1. Frank Mittelbach, Michel Goossens, Johannes Braams, DavidCarlisle, Chris Rowley, ―LaTeX Addison Wesley.

MCA 402
C Programming

Course Objectives:

The course is designed to provide knowledge of C language. Students will be able to develop logics which will help them to create programs, applications in C. Also by learning the basic programming constructs they can easily switch over to any other language in future. Student will learn the fundamental programming concepts and methodologies

Course Level Learning Outcomes:

Upon successful completion of the course students will be able to:

- Design programs connecting decision structures, loops and functions.
- Explain the difference between call by value and call by address.
- Understand the dynamic behavior of memory by the use of pointers.
- Understand theconcepts like arrays, strings, structure, and union.
- Understand the concept of files handling in C.

UNIT-I

Overview of C- General Structure of C Program, C compilers, Editing, Compiling & , Running of a C program Data types, Constants and Variables, Operators and expressions, Storage Classes, Different types of expressions and their Evaluation, Conditional Expression, Assignment statement, Enumerated data type, Redefining/ Creating data types, Library functions, Type casting.Input/Output-Unformatted and formatted I/O Functions.

UNIT-II

Control Statements- Decision making using if, if-else, elseif and switch statements, Looping using for, while and do-while statements, Transferring Program controlling break and continuestatements, Programming examples to illustrate the use of these control statements.

Functions- Defining a function, Local variables, return statement, invoking a Function, specifying and passing arguments to a function, Functions returning non Integer, External, static, and register variable, block structure, initialization and recursion.

UNIT-III

Array & strings- Introduction to arrays, Declaring arrays, Initializing, arrays, Processing arrays, Pointers to arrays, Passing arrays as arguments to functions, Introduction to strings, Pointers to strings, Passing strings and Arrays of strings as arguments to a function, Programming examples to illustrate the use of arrays and strings.

Pointers- Definition, Need of pointers, declaring Pointers, Accessing Values via Pointers, Pointer arithmetic, Types of pointers, Programming examples to illustrate the use of pointers.

UNIT-IV

Structures- Declaring a structure type, Declaring Variables of structure type, Initializing Structures, Accessing Elements of structures, arrays of structures, nested structures, Pointers to structures Programming examples to illustrate the use of Structures. File Handling.

Text Books:

1. E. Balagurusamy, "Programming in ANSI C", 8E ,Tata McGraw Hill.

Reference Books:

1. R S Salaria, Application in C, Khanna book publishing.
2. YashwantKanetakar, "Let us C" BPB.
3. Kerninghan B.W. & Ritchie D.M. "The C Programming Language" Prentice-Hall.
4. Mullish Cooper, "The Spirit of C" Jaico Publishing House.
5. Byron Gottfried, "Programming with C", Schaum's Outlines, Tata McGraw Hill.
6. Herbert Schildt, C: The complete reference, Tata mcCgraw hill.

MCA-401
Fundamental of ICT

Course Objective:

This course seeks to provide students with a thorough overview of why computers are necessary in business, education, and humanity. It also provides a fundamental understanding for non-computer science students to become acquainted with courses such as MCA.

Course Outcomes

After completing the course, the student should be able to:

- Be familiar with the fundamental architecture of computer systems.
- Know the fundamentals of number systems such as binary, octal, and hexadecimal.
- Know how to distinguish between different operating systems such as Windows and Linux

UNIT-I

Introduction: Computer, Data Processing, Computer System Characteristics, Evolution ofComputers, Capabilities and Limitations, Generations of computers, Block diagram of computer,Basic components of a computer system- Input unit, Output unit, Storage unit, ALU, Control unit, Central Processing unit; Number Systems- Non-positional number system, Positional number system, Decimal Number system, Binary number system, Octal number system, Hexadecimal number system.

UNIT-II

Memory: Main memory organization, Main memory capacity, RAM, ROM, PROM, EPROM,Cache Memory, Secondary storage devices: Sequential access devices- Magnetic tape; Direct access devices-Magnetic disks, Floppy disks, Optical disks, Types of Optical disks: CD-ROM, CDR, CD-RW, DVD.

Input devices: Keyboard, Pointing Devices-Mouse, Touch screens, Joystick, Electronic pen,Trackball, Scanning devices: Optical Scanners, OCR, OMR, Bar code reader, MICR, Electroniccard reader, Image capturing devices, Digital cameras.

Output devices: Monitors- CRT, LCD, Printers-Dot matrix, Inkjet, Laser; Plotters, Screen imageprojector.

UNIT-III

Introduction: Software, Relationship between Hardware and Software, Types of Software-System Software, Application Software; System Software-Operating System, Utility Program;Programming Languages-Machine, Assembly, High Level; Assembler, Compiler, Interpreter.

UNIT-IV

Data Communication & Computer Networks, Basic elements of a communication system, DataTransmission modes-Simplex, Half duplex, Full duplex; Data Transmission speed-Narrowband,Voice band, Broadband; Data Transmission media-Twisted Pair Wire, Coaxial cable, Optical fibers; Modems, Types of Network-LAN, WAN, MAN; Internet, World Wide Web, Web Browsers.

Text Book:

1. Pradeep K. Sinha, PritiSinha, “Computer Fundamentals”, 6E ,BPB Publications.

Reference Books:

1. Rajaraman, V., “Fundamental of Computers”, Fifth Edition, Prentice Hall India, New Delhi.
2. E. Balagurusamy, “Introduction to Computers (Special Indian Edition)”, Tata McGraw Hill.