ANNEXURE I

Krishna Kanta Handiqui State Open University

Course Curriculum of MSc in Information Technology (M.Sc.IT) Programme

Credit Distribution for M.Sc.IT Programme

SL No	Nature of Courses	Total no. of Courses	Credit	Total Credits
1	DSC (Discipline Specific Core)	12	4	48
2	DSE (Discipline Specific Elective)	04	4	16
3	AEC (Ability Enhancement Course)	02	4	08
4	VAC (Value Added Course)	02	2	04
5	Seminar	01	4	04
6	Project/Dissertation	01	8	08
	Total	22		**88**

Course Distribution for M.Sc.IT Programme

Semester	DSC	DSE	AEC	VAC	Dissertation/ Seminar	Semester wise credits
I	DSC 1, DSC 2, DSC 3	DSE 1	AEC 1	VAC 1		22
II	DSC 4, DSC 5, DSC 6	DSE 2		VAC 2	Seminar	22
III	DSC 7, DSC 8, DSC 9	DSE 3	AEC 2			20
IV	DSC 10, DSC 11, DSC 12	DSE 4			Project/Dissertation	24
Total Credits	48	16	8	4	12	**88**

Semester-wise Course and Credit Distribution

Semester	**DSC** (12 Courses, Credit 4 each)	**DSE** (4 Courses, Credit 4 each)	**AEC** (2 Courses, Credit 4 each)	**VAC** (2 Courses, Credit 2 each)	**Dissertation/ Seminar** (Seminar: 4 Credit, Project: 8 Credit)	**Total Credits**
I	Computer Fundamentals and Digital Techniques Computer Programming using C Fundamentals of Networking and Web Technology	PC Software OR Open Source Software	AEC 1 (any one) (SWAYAM Course on *Digital Forensics*) and other options mentioned in Annexure V)	VAC 1 (any one course from Annexure V)		**22**
II	Computer Organization and Architecture Data Structure through C Language Fundamentals of Database Management System	Linux System Administration OR System Analysis and Design		VAC 2 (any one) (SWAYAM Course on *Academic Writing* and other options mentioned in Annexure V)	Seminar	**22**
III	Introduction to Discrete Mathematics and Formal Languages and Automata Object-Oriented Programming through C++ Operating System	Programming in Java OR Computer Graphics	AEC 2 (any one course from Annexure V)			**20**
IV	Artificial Intelligence and Machine Learning Introduction to Cloud Computing Programming in python	Mobile Application Development OR Image Processing			Project/ Dissertation	**24**
Total Credit	**48**	**16**	**8**	**4**	**12**	**88**

Marks Distribution of Courses for MSc in Information Technology

Semester	Course	Whether DSC/DSE/AEC/VAC	Marks	Credit
1st	Computer Fundamentals and Digital Techniques	DSC	30F+70S	4
	Computer Programming using C	DSC	20F+30P+50S	4
	Fundamentals of Networking and Web Technology	DSC	20F+30P+50S	4
	(Choose any one) PC Software Open Source Software	DSE	20F+30P+50S	4
	AEC 1	AEC	30F+70S	4
	VAC 1	VAC	30F+70S	2
2nd	Computer Organization andArchitecture	DSC	30F+70S	4
	Data Structure through C Language	DSC	20F+30P+50S	4
	Fundamentals of Database Management System	DSC	20F+30P+50S	4
	(Choose any one) Linux System Administration System Analysis and Design	DSE	20F+30P+50S	4
	VAC 2	VAC	30F+70S	2
	Seminar	Dissertation/Seminar	100	4
3rd	Introduction to Discrete Mathematics and Formal Languagesand Automata	DSC	30F+70S	4
	Object-Oriented Programming through C++	DSC	20F+30P+50S	4
	Operating System	DSC	30F+70S	4
	(Choose any one) Programming in Java Computer Graphics	DSE	20F+30P+50S 30F+70S	4
	AEC 2 (Cyber Security)	AEC	30F+70S	4
4th	Artificial Intelligence and Machine Learning	DSC	30F+70S	4
	Introduction to Cloud Computing	DSC	30F+70S	4
	Programming in Python	DSC	20F+30P+50S	4
	(Choose any one) Mobile Application Development Image Processing	DSE	20F+30P+50S 20F+30P+50S	4
	Major Project		200	8
				88

N.B.:

F = Formative Assessment (i.e. Internal Assessment)
P = Practical
S = Summative Assessment (i.e. End Term Examination)
1 credit = 30 hours of learning
3 Credit = 30 X 4 =120 hours of learning

ANNEXURE II

M.SC.IT DETAILED COURSE WISE SYLLABUS

Syllabus of Discipline Specific Core (DSC) Courses

Semester I: Computer Fundamentals and Digital Techniques

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 30, Summative: 70)

Course Objectives:

- To introduce students to the basic concepts of computer systems and their components.
- Familiarization with digital logic and Boolean algebra, including the basic gates, Boolean expressions, and truth tables.
- To familiarize students with the principles of digital electronics and their applications in computer systems.

Course Outcomes:

On completion of this course, the learners will be able to

- understand the fundamental concepts of computer organization and architecture, including hardware and software components, input/output devices, memory, and storage systems.
- Understand the basics of data representation, including binary, decimal, and hexadecimal numbering systems.
- gain knowledge of digital logic circuits and the design and analysis of combinational and sequential circuits.

Unit 1: Introduction to Computer
Block Diagram of Computer, Evolution, Generations, Classification and its Application

Unit 2: Basic Components of Computer
Concept of Bit and Byte; CPU, ALU, CU; Computer Memory: Primary and Secondary; Input/Output Devices; Buses: Address, Data and Control.

Unit 3: Introduction to Computer Security
Computer Virus, Worm, Spyware, Malware, Trojan horse, Antivirus Software

Unit 4: Threats and Legal Framework
Cyber Crimes against Individuals, Institution and State, Hacking, Digital Forgery, Cyber

Stalking/Harassment, Cyber Pornography, Identity Theft & Fraud, Cyber terrorism, Cyber Defamation, Different offences under IT Act, 2000

Unit 5: Introduction to Number Systems
Decimal, Binary, Hexadecimal and Octal number system, Number system conversion

Unit 6: Binary Arithmetic
Complement: r's and (r-1)'s complement, Binary addition, Binary subtraction, Binary Multiplication, Binary division.

Unit 7: Data Representation
Fixed Point representation and Floating point representation

Unit 8: Code Conversion Technique
Gray code, BCD, Code conversion technique, ASCII, EBCDIC, Unicode, ISO

Unit 9: Boolean algebra
Introduction, Properties of Boolean Identities, Properties of Positive and Negative Logic, De-Morgan‟s Theorem, Application of De-Morgan‟s theorem.

Unit 10: Logic Gates
Logic Gates: AND, OR, NOT, NAND, NOR, XOR; Conversion of the Logic Gates, NAND and NOR Implementation of Basic Gates.

Unit 11: Reduction Techniques
Boolean Expression, Simplification of Boolean expression by Algebraic method and Karnaugh Map, SOP, POS, Standard and Canonical form, Conversion of SOP to POS and vice-versa, Reduction upto 3-variables.

Unit 12: Combinational Circuit
Definition and Properties of Combinational Circuit, Introduction to Multiplexer, 4-to-1 multiplexer, Demultiplexer, Encoder, Decoder,

Unit 13: Binary Arithmetic Circuits
Half adder, Full adder, Binary Half subtractor, Full subtractor

Unit 14: Sequential Circuit
Definition and Properties of Sequential Circuit, Block Diagram of Sequential Circuit, Types of Sequential Circuit, Introduction to Flip-Flop and its different types: RS, JK; Master-Slave JK Flip-Flop.

Unit 15: Registers
Types of Shift Registers: SISO, PISO, SIPO, PIPO;

Unit 16: Counters
Synchronous and Asynchronous Counter, Limitations of Asynchronous Counter.

References:

1. Mano, M. M., & Kime, C. R. (2017). Logic and Computer Design Fundamentals. Pearson.

2. Floyd, T. L. (2015). Digital Fundamentals. Pearson.

Semester I: Computer Programming using C

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 20, Practical: 30, Summative: 50)

Course Objectives:

- Develop proficiency in programming using the C language.
- Understand basic programming concepts such as data types, variables, control structures, functions, and arrays.
- Acquire skills to design and implement efficient algorithms and programs using C language.
- Learn to use C programming to solve real-world problems and challenges.

Course Outcomes:

On completion of this course, the learners will be able to

- write and debug C programs using programming constructs like variables, data types, control structures, functions, arrays, and pointers.
- design and implement programs that meet specific requirements using C programming techniques, algorithms, and data structures.
- apply their skills to a wide range of real-world problems. This includes the ability to analyze a problem, break it down into smaller parts, and develop a solution using programming concepts.

Unit 1: Introductory Concepts
Basic definition of Pseudo Code, algorithm, flowchart, program

Unit 2: Elements of C Programming
Characters used in C, Identifiers, Keywords, Tokens, Constants, Variables, Types of C variables,Receiving input and output

Unit 3: Variables and Data types
Integer, character floating point and string; Initialization of variable during declarations; Symbolic Constants, type conversion in assignment

Unit 4: Operators and Expressions

Expression in C, Different types of operators: Arithmetic, Relational and Logical, Assignment, Conditional, Increment and decrement, Bitwise, Comma and other operator (size of, period etc). Precedence and associatively of operators, type casting

Unit 5: I/O Functions

Header Files (stdio, conio), Formatted Input/Output Functions (scanf, printf), Escape Sequences, Character Input/Output Functions (getch, getchar, putchar, gets, puts, getche, clrscr)

Unit 6: Preprocessor Directives

Features of C preprocessor, Macro expansion, Macros with arguments, #if and #elif directives,

Unit 7: Conditional Statements

Conditional Statement- if, if- else, nested if-else, switch-case; break, continue, goto

Unit 8: Loop Control Structures

Concept of Loops, Types of loop: while, do-while, for; nested loops

Unit 9: Storage Class

Automatic, External, Static, Register, Scope and lifetime of variables, Macro, Preprocessor directive

Unit 10: Arrays

Array, Array Declaration, 1-Dimensional array, 2-Dimensional array

Unit 11: Strings

String, String Handling Functions: strlen(), strcmp(), strcpy(), strrev(), strcat(), etc

Unit 12: Functions

Function, Function declaration, Function definition, Function call, Formal and Actual parameter,Recursive function

Unit 13: Pointers

Pointer, Pointer declaration, Passing pointer to a function, Pointer and one-dimensional arrays, Dynamic memory allocation

Unit 14: Structures and Union

Structure Declarations, Definitions, Defining your typedef, Array of Structure, Pointer to Structure. Union Declaration, Definition, Declaration, Uses

Unit 15: File Handling

Concept of File, File Pointer, File Opening in various modes, closing a file, reading and writing on files, Formatted Input/Output, fseek(), ftell(), rewind()

References:

1. Kanetkar, Y. (2018). Let Us C (16th ed.). Delhi: BPB Publications.
2. Balaguruswamy, E. (2019). Programming in ANSI C (7th ed.). Delhi: Tata McGraw-Hill Education Private Limited.

SEMESTER I: Fundamentals of Networking and Web Technology

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 20, Practical: 30, Summative: 50)

Course Objectives

- To provide a comprehensive understanding of the fundamental concepts and principles of networking, including network architectures, protocols, topologies, and communication models.
- To develop an understanding of the different types of networks, their architectures, and their components.
- To introduce learners to web technologies and their applications, such as HTML and CSS and enable them to design and develop simple web applications.

Course Outcomes

On completion of this course, the learners will be able to

- understand the fundamental concepts and principles of computer networking and web technologies, including the internet, data transmission, network architecture, and communication protocols.
- analyse network architectures and design principles, such as the OSI model and TCP/IP protocol suite.
- gain hands-on experience with web technologies and programming languages such as HTML and CSS.

Unit 1: Introduction to Computer Network
Goals of Computer Network, Types of Computer Network: LAN, MAN, WAN, LAN Transmission Methods, Peer-to-Peer LANs, Broadcast and Point-to-Point Networks, Connection-Oriented and Connection-Less Services.

Unit 2: Network Topology
Definition of Network Topology, Types of Network Topology: Bus, Ring, Star, Mesh, Tree.

Unit 3: Data Communication Modes
Introduction to Signals and Systems, Types of Communication (Analog, Digital), Modes of Communication (Simplex, Half-Duplex, Full-Duplex), Modulation.

Unit 4: Network Devices
Network Interface Card (NIC), Modem, Switch, Router, Gateway.

Unit 5: Data Transmission
Parallel Transmission, Serial Transmission, Bandwidth, Baud length, Switching Techniques: Circuit Switching, Message Switching, and Packet Switching.

Unit 6: Network Models
ISO-OSI Reference Model: Functions of each Layer; Various Terminology used in Computer Network; Connection-Oriented and Connectionless Services, TCP/IP Reference Model, Comparison of ISO-OSI and TCP/IP Model.

Unit 7: Transmission Media
Transmission Medium, Guided Media: Coaxial Cable, Twisted Pair, Fiber Optics Cable: Unguided Media: Radio Waves, Bluetooth, Infrared, Microwave, Satellite, Wireless LANs (IEEE 802.11).

Unit 8: Internet
Definition of Internet, Internet Architecture, Peer-to-Peer, Client-Server, Accessing Technique, Internet Service Providers, Organization of Internet, Internet Protocol Suite, IP Address, DomainName System, Uniform Resource Locators (URL), Socket, Application of Internet.

Unit 9: Word Wide Web
Web Page, Web Browsers, Web Server, Web Search Engines, Web Cookies, Technologies in WWW.

Unit 10: Introduction to HTML
Basics of HTML, HTML Editor, HTML Tags, Document Structure Tags, Formatting Tags, Hyperlink and Image tags

Unit 11: HTML Lists and Tables
HTML Lists: Unordered, Ordered, Definition Lists; HTML Table tags;

Unit 12: Advanced HTML Tags
Frame tags, HTML Form, Additional Advanced HTML Tags.

Unit 13: Introduction of XML
Introduction to XML, Comparison of HTML and XML, XML Basics, Introduction to DTD

Unit 14: Introduction to CSS
Cascading Style Sheets (CSS), Inline Style, Embedded Style, External Style Sheet, Imported Style Sheet, Ruleset, @ rule, Class Selector, ID Selector, Contextual Selector, Attribute Selector,CSS Properties – background properties, text properties, and border properties

Unit 15: Dynamic HTML
Dynamic HTML, Document Object Model, Features of DHTML

References:

1. Forouzan, B. A., & Fegan, S. C. (2019). Data communications and networking. McGraw-Hill Education.
2. https://www.w3schools.com/

SEMESTER II: Computer Organization and Architecture

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 30, Summative: 70)

Course Objectives

- To understand the basic principles and components of computer organization and architecture, including instruction sets, memory hierarchy, input/output devices, and system bus.
- To develop an understanding of how data is processed in a computer system, including how instructions are fetched, decoded, and executed, and how data is stored and retrieved from memory.
- To explore the advanced concepts of computer architecture, such as parallel processing, multiprocessor systems, and distributed computing.

Course Outcomes

On completion of this course, the learners will be able to

- describe and analyze the fundamental concepts of computer organization and architecture, including digital logic design, instruction set architecture, and memory hierarchy.
- explain how various computer components, such as processors, memory, and input/output devices, interact and communicate with each other to execute instructions and perform tasks.
- Knowledge of the different types of computer memory, their characteristics and uses, and the design and implementation of memory subsystems.

Unit 1: Basic organization of the computer
Basic organization of the computer and block level description of the functional units from program execution point of view; Fetch, decode and execute cycle

Unit 2: Digital Components
Half adder, Full Adder, Half subtractor, Full subtractor, Coder-Decoder, Multiplexer, De-multiplexer, Magnitude Comparator, Flip-Flops, Counter, Register

Unit 3: Data Representation
Data representation, computer arithmetic and their implementation; control and data path, data path components, design of ALU and data path, control unit design.

Unit 4: Computer Arithmetic
Integer representation, sign magnitude representation, twos complement representation, integer arithmetic

Unit 5: Instruction Sets
Elements of a machine instruction, instruction representation, Simple instruction format, Instruction types, number of addresses, Types of operands, Types of operations. Different Instruction Formats, Instruction Types, Instruction Execution, Assembly language notation

Unit 6: Addressing modes
Addressing: Immediate, Direct, Indirect, Registrar, Registrar indirect, Relative Index.

Unit 7: Input-Output Organization
Different I/O techniques (Programmed I/O, Interrupt-Driven I/O), DMA (Direct Memory Access), I/O Processors

Unit 8: Introduction to Cache and Virtual Memory
Memory Hierarchy, Semiconductor memories, internal organization of typical RAM and ROM Memory, Switches, Cache memory, Cache memory access techniques; Mapping functions, Virtual memory, Locality of reference, Paging, Cache Coherence Problem

Unit 9: Memory and I/O access
Memory Read Write operations, Concept of handshaking, Polling Techniques (Serial and Hub Polling) and Interrupt driven I/O, Priority and Daisy Chaining Technique, Introduction to Memory Mapping.

Unit 10: Memory and I/O Interfacing
I/O processor, Priority Encoder, Device Scheduler, Interfacing with the I/O Devices, keyboard, printer and display interfaces

Unit 11: External Memory
Magnetic Disk, Magnetic read write operation, sector, track, inter track and inter sector gap, cylinder, fixed head disk, seek time, access time, transfer time, rotational delay, RAID, Optical memory, Magnetic tap.

Unit 12: Processor structure and Function

Processor organization: Fetch instruction, interpret instruction, fetch data, process data, write data, Registrar organization, Control and status registrar, Instruction cycle, Instruction pipelining(definition only)

Unit 13: Introduction to Parallel Processing
Introduction to Pipelining and Basics of Parallel Processing, Scalable Architecture

Unit 14: Introduction to CISC and RISC Architecture
CISC (Complex Instruction Set Computers), RISC (Reduced Instruction Set Computers), Examples of CICS and RISC

Unit 15: Parallel Architectures
SISD, SIMD, MISD, MIMD, Scalar, Flynn's Classification of Computer Architecture, Vector, superscalar and pipelined processor, Pipelining, Instruction pipeline, pipeline bubbles, Hazards: -resource conflicts, data dependency, branch difficulty. Vector computing, arithmetic pipeline, vector and scalar register, chaining, scatter gather operations, vector-register processor, memory-memory vector processor. Array processor

References:

1. Stallings, W. (2018). Computer Organization and Architecture: Designing for Performance. Pearson Education.
2. Mano, M. M. (2019). Computer System Architecture. Pearson Education

SEMESTER II: Data Structure through C Language

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 20, Practical: 30, Summative: 50)

Course objectives:

- Understand the fundamental concepts of data structures and their implementation using the C programming language.
- Develop the ability to create, implement and analyse different types of data structures including arrays, linked lists, stacks, queues, trees, and graphs using C language.
- Understand the various techniques for sorting and searching algorithms and their implementation in C.
- Develop skills to choose the appropriate data structure for solving a given problem, and implement them efficiently using C programming language.

Course outcomes:

On completion of this course, the learners will be able to

- understand of basic data structures like arrays, linked lists, queues, stacks, and trees. The learners should be able to create, manipulate and implement these data structures using the C programming language.
- identify and analyze algorithms used for sorting, searching, and manipulating data structures. They should be able to evaluate the efficiency of algorithms in terms of time and space complexity.
- design efficient algorithms to solve real-world problems using the appropriate data structures. They should be able to understand how to optimize code and reduce time complexity to solve large-scale problems.

Unit 1: Introduction to Data Structure

Basic concept of data, data type, Elementary structure, Arrays: Types, memory representation, address translation functions for one & two dimensional arrays and different examples.

Unit 2: Algorithms

Complexity, time-Space, Asymptotic Notation

Unit 3: Linked List

Introduction to Linked List, representation of single linked list, linked list operations: Insertion into a linked list, deletion a linked list, searching and traversal of elements and their comparative studies with implementations using array structure.

Unit 4: Stack

Definitions, representation using array and linked list structure, applications of stack.

Unit 5: Queue

Definitions, representation using array, linked representation of queues, application of queue.

Unit 6: Searching

Linear and Binary search techniques, Advantages and disadvantages, Analysis of Linear and Binary search

Unit 7: Sorting

Sorting algorithms (Complexity, advantages and disadvantage, implementation), bubble sort, insertion sort, selection sort, quick sort

Unit 8: Trees

Definition and implementation: Binary Tree, Tree traversal algorithms (inorder, preorder, postorder), postfix, prefix notations; Binary Search Tree: Searching in BST, insertion and deletion in BST.

Unit 9: Dictionary ADT

Search trees, balancing of search trees – AVL trees, Red-Black trees, multi way search trees, 2-3 trees, splay trees, Insertion and Deletion in each of the above data structures, Hashing.

Unit 10: Advance Sorting and Selection Techniques

Heap sort, Shell sort, sorting in linear time, Counting sort, Radix sort. Medians and order Statistics Selection and Adversary arguments. Lower bound on sorting

Unit 11: Priority Queue ADT

Heaps-extended priority queue, min(max) heaps, binomial heap, fibonacci heap and its amortizedanalysis.

Unit 12: Partition ADT

Union-find algorithms through weighted merge and path compression

Unit 13: Data Structure for external storage operations

B-tree, insertion and deletion in B-trees, external sorting, B^+ tree

Unit 14: Red Black Tree

Definition, properties, Creation, Insertion, left and right rotation, balancing red black tree, deleting node from red black tree

Unit 15: Graph

Introduction to Graph, Graph representation: adjacency matrix, adjacency list, Traversal of graph: depth first search and breadth first search.

References:

1. Kanetkar, Y. (2019). Data Structures Through C. BPB Publications.
2. Balaguruswamy, E. (2020). Data Structures Through C. Tata McGraw-Hill Education.

SEMESTER II: Fundamentals of Database Management System

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 20, Practical: 30, Summative: 50)

Course objectives:

- To develop a comprehensive understanding of the fundamental concepts, principles, and practices of database management systems, including data modeling, database design, and query optimization.
- To develop skills in designing and implementing database systems using industry-standard database management software.
- To understand the role of database management systems in modern organizations, and the importance of data security, privacy, and compliance.
- To be able to effectively manage and maintain database systems, including troubleshooting, backup and recovery, and performance tuning.

Course outcomes:

On completion of this course, the learners will be able to

- understand the concepts and principles of database management systems. This includes data modeling, normalization, and schema design.
- learn how to design and implement databases, including the creation of tables, indexes, and other database objects. They will also learn how to write SQL queries to retrieve and manipulate data.
- gain knowledge of managing database security and administration, including user authentication, backup and recovery, and monitoring of database performance.

Unit 1: File Structure and Organization
Data and Information, Concept of Field, Key Field; Records and its types, Fixed length records and Variable length records; Files, operation on files, Primary file organization.

Unit 2: Database Management System
Definition of DBMS, File processing system vs. DBMS, Advantages and Disadvantages of

DBMS, Database Architecture, Data Independence, Data Dictionary, DBMS Language, Database Administrator.

Unit 3: Data Models
Data Models: Object Based Logical Model, Record Base Logical Model, Relational Model, Network Model, Hierarchical Model.

Unit 4: Entity-Relationship Model
Entity Set, Attribute, Relationship Set, Entity Relationship Diagram (ERD), Extended features ofERD.

Unit 5: Relational Databases
Relational data model; Terms: Relation, Tuple, Attribute, Cardinality, Degree, Domain; Keys : Super Key, Candidate Key, Primary Key, Foreign Key;

Unit 6: Relational Algebra
Operations: Select, Project, Union, Difference, Intersection, Cartesian Product, Natural join.

Unit 7: SQL (Part I)
Introduction of SQL, characteristic of SQL, Basic Structure, DDL Commands, DML, DQL, SELECT Statement, WHERE Clause, Useful Relational Operators, Aggregate Functions, SUM Function, AVG Function.

Unit 8: SQL (Part II)
Compound Conditions and Logical Operators, AND Operator, OR Operator, Combining AND and OR Operators, IN Operator, BETWEEN Operator, NOT Operator, Order of Precedence for Logical Operators, LIKE Operator, Concatenation Operator, Alias Column Names, ORDER BY Clause, Handling NULL Values, DISTINCT Clause

Unit 9: Normalization of Database
Introduction to Normalization, Anomalies of un-normalized Database, Normalization of Database: 1NF, 2NF, 3 NF, BCNF.

Unit 10: Database Design
Design of Relational Database, Functional Dependency Diagram, Dependency-preserving property, lossless join property, algorithms to ensure dependency -preserving property and lossless join property.

Unit 11: System Implementation Techniques
Query processing and optimization- translation between SQL queries and relational algebra;

Unit 12: Transaction Processing
Transaction and System concepts, Desirable properties, Schedules and Recoverability

Unit 13: Concurrency Control
Locking Techniques, Concurrency Control based on timestamp ordering, Multiversion Concurrency Control Techniques

Unit 14: Database Recovery
Concepts and techniques, recovery in multi database systems

Unit 15: **Security and Authentication**
Issues, access control techniques, introduction to multilevel security

References:

1. Navathe, R. (2018). Fundamentals of database systems. Pearson.
2. Chopra, R. (2019). Database Management System (3rd ed.). New Delhi: Pearson Education.

SEMESTER III: Introduction to Discrete Mathematics and Formal Languages and Automata

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative:30, Summative: 70)

Course objectives:

- To understand the fundamental principles of logic and set theory and their application in discrete mathematics.
- To learn the basics of graph theory, including properties of graphs and algorithms for analyzing graphs.
- To provide students with an understanding of the fundamental concepts and principles underlying formal languages and automata, including regular expressions, finite automata, and context-free grammars.

Course outcomes:

On completion of this course, the learners will be able to

- apply mathematical concepts and techniques to analyze discrete structures and problems, including graphs, trees, relations, and functions.
- understand the structure and properties of formal languages and automata, including regular expressions and context-free grammars, and apply this understanding to problems in computer science and Information Technology.
- apply the principles of discrete mathematics and formal languages to the design and analysis of algorithms, and demonstrate proficiency in algorithmic analysis.

PART I: Discrete Mathematics

Unit 1: Sets, Relations and Functions
Sets – the Empty Set, Finite and Infinite Set, Equal and Equivalent set, Subsets, Power set, Universal set, Venn diagram, Complement of a set, set operations; Concept of relation: identity and inverse relation, types of relation, equivalence relation; Concept of function: identity and constant function, types of function.

Unit 2: Lattices
Lattices as partially ordered sets, their properties. Lattices and algebraic systems; Sub lattices, direct products and homomorphism. Some special lattices for example complimented and distributive lattices.

Unit 3: Propositional Logic
Statements, logical connectives, truth tables; Tautologies, contradictions, logical equivalence,Applications to everyday reasoning

Unit 4: Boolean Algebra
Boolean Algebra, Relation of Predicate Calculus to Boolean algebra

Unit 5: Counting Principles
The Pigeonhole principle - counting; Permutation and Combination: Definition of Permutation and combination, Simple application of permutation and combination, Principle of Inclusion-Exclusion

Unit 6: Basic Algebraic Structure
Binary operations, identity and inverse of an element, group, subgroup, coset, cyclic group,normal subgroup, quotient group; Ring, Commutative Ring, Integral domain and Field

Unit 7: Graph Theory
Basic concepts- finite and infinite graphs, incidence and degree, isolated and pendant vertices, null graph; Paths and Circuits- isomorphism, subgraphs, walks, connected and disconnected graphs and components, Euler graphs, Bi-partite graphs, Hamiltonian paths and circuits;

Unit 8: Trees
Trees, Properties of trees, distance and centers, rooted and binary trees, on counting trees, spanning, fundamental circuits, spanning trees in weighted graphs; Cut-sets- properties, connectivity and separability, network flows; Matrix representation of graphs- incidence matrix, submatrices, circuit matrix, cut-set matrix, path matrix, adjacency matrix;

PART II: Formal Languages and Automata

Unit 9: Mathematical preliminaries
Set, Relations and Functions; Properties of Relations, Closure of Relations, Symbols and Alphabets, Strings and their properties, Languages, Principle of Induction.

Unit 10: The Theory of Automata
Definition of Automaton, Finite Automata, Transition Systems, Properties of Transition Functions, Acceptability of a String by a Finite Automata, Finite Automata with Epsilon-transitions;

Unit 11: Types of Finite Automata

Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA); The Equivalence of DFA and NDFA; Minimization of Finite Automata; Application of Finite automata;

Unit 12: Formal languages
Basic Definition, Definition of a Grammar, Derivation and language generated by a grammar, Chomsky Classification of Languages;

Unit 13: Regular Expressions and Regular Grammars
Regular Set and Regular Grammars; Closure Properties of Regular Languages ; Regular Expressions; Relationship between the Finite Automata and Regular Expressions; The Pigeon-hole principle; Applications of Regular Expressions;

Unit 14: Context-Free Grammars and Languages
Context-Free grammars; Derivation tree, Parse trees; Applications; Ambiguity in Grammars and Languages

Unit 15: Properties of Context-Free Languages
Normal forms for CFGs; The pumping lemma for CFGs; Closure properties of CFL

References:

1. Hopcroft, Automata Theory Language and Computation. Pearson.
2. Bawa, N.P.S. Essesntials of Discrete Mathematics. Pragya Publications

SEMESTER III: Object-Oriented Programming through C++

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 20, Practical: 30, Summative: 50)

Course objectives:

- The course aims to provide a strong foundation in the concepts of object-oriented programming. This includes topics such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction.
- The course also focuses on developing programming skills in C++, a widely used programming language in the industry. Students will learn about the syntax, data types, control structures, functions, and libraries in C++.
- The course aims to equip students with the skills to apply object-oriented concepts in real-world problems. This includes designing and implementing software solutions using object-oriented programming techniques.
- By the end of the course, students should be able to build software applications using object-oriented programming in C++. They should be able to analyze problems, design software solutions, and implement them using C++ programming concepts.

Course outcomes:

On completion of this course, the learners will be able to

- design and implement object-oriented programs in C++ using classes and objects.
- apply inheritance, polymorphism, and encapsulation concepts to create reusable and extensible code.
- use advanced programming techniques such as templates, exceptions, and streams to build robust and efficient C++ programs.

Unit 1: Introduction to Object-Oriented Programming
Basic concept of OOP, Comparison of Procedural Programming and OOP, Benefits of OOP, C++ compilation, Abstraction, Encapsulation, Inheritance, Polymorphism, Difference between Cand C++

Unit 2: Elements of C++ Language
Tokens and identifiers: Character set and symbols, Keywords, C++ identifiers. Variables and constants: Integers & characters, Constants and symbolic constants, Dynamic initialization of variables, Reference variables, Basic data types in C++, Streams in C++, scope resolution operator

Unit 3: Operators and expressions
Operators, Types of Operators in C++, Precedence and Associativity, Manipulators, Enumerateddata type, storage classes

Unit 4: Decision and Control Structures
if statement, if-else statement, switch statements, Loop: while, do-while, for; Jump statements :break, continue, goto

Unit 5: Array and Structure
Arrays, structure, unions, Runtime memory management: new and delete operator

Unit 6: Pointer
Introduction, Address operator, pointer variable, pointer definition, pointers and parameter passing, void pointer, pointer arithmetic, Runtime memory management: new and delete operator, Pointers to pointer, Passing address of a pointer, Array of Pointers, Pointers to functions

Unit 7: Functions
main() function, components of function : prototype, function call, definition, parameter; passingarguments; types of function, inline function, function overloading

Unit 8: Introduction to Classes and Objects
Classes in C++, class declaration, declaring objects, Defining Member functions, Inline memberfunction, Array of objects, Objects as function argument, Static data member and mmber function, Friend function and friend class.

Unit 9: Constructors and Destructors
Constructors, Instantiation of objects, Default constructor, Parameterized constructor, Copy constructor and its use, Destructors, Constraints on constructors and destructors, Dynamic initialization of objects.

Unit 10: Operator Overloading
Overloading unary operators: Operator keyword, arguments and return value; Overloading Unary and binary operators: arithmetic operators, manipulation of strings using operators, Typeconversions.

Unit 11: Inheritance
Derived class and base class: Defining a derived class, Accessing the base class member, Inheritance: multilevel, multiple, hierarchical, hybrid; Virtual base class, Abstract class.

Unit 12: Virtual Functions and Polymorphism
Virtual functions, Pure virtual functions; Polymorphism, Categorization of polymorphism techniques: Compile time polymorphism, Run time polymorphism.

Unit 13: File Handling
File classes, Opening and Closing a file, File modes, Manipulation of file pointers, Functions forI/O operations

Unit 14: Templates
Introduction, Function template, Function template overloading, Class templates, Template arguments

Unit 15: Object Oriented Design
Object Oriented Design Approaches, Object Modeling Techniques (OMT) tools: Object Model,Dynamic Model, and Functional Model. (Object Diagram, State Diagram, and DFD), Phases ofObject-Oriented Development: Object Analysis, System Design, Object Design

References:

1. Balaguruswami, E. (2020). Object-Oriented Programming through C++ (6th ed.). McGraw Hill Education.
2. Lafore, R. (2002). Object-Oriented Programming in C++. Sams Publishing.

SEMESTER III: Operating System

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative:30, Summative: 70)

Course objectives:

- To provide an understanding of the fundamentals of operating systems, including the role of an operating system, its structure and components, and how it interacts with hardware and software.
- To explore different types of operating systems, including single user, multi programmed, and multi-user systems, and understand their applications and advantages.
- To introduce the concepts of process and thread management, memory management, file

systems, and input/output operations, and how they are implemented in different operating systems.

Course outcomes:

On completion of this course, the learners will be able to

- understand the basic concepts and principles of operating systems, including process management, memory management, file systems, and device management.
- evaluate the performance of operating systems, including how to optimize system resources, enhance system security, and troubleshoot common operating system problems.
- design and implement operating system components and features, including process scheduling algorithms, memory allocation policies, file system design, and device drivers.

Unit 1: Review of Computer Organization
Major Subsystems, I/O Organization, Memory Organization, Bus Organization.

Unit 2: Introduction to Operating System
Definition of Operating System, Batch System, Multi-Programmed System, Time-Sharing System, Single User Operating System, Multi-User Operating System.

Unit 3: Processes
Process: process models, process hierarchies, process states; Threads: use, design issues ofthread; Types and Application.

Unit 4: Interprocess Communication
Interprocess communication, race conditions, critical-sections, mutual exclusion, solution to race condition, disabling interrupt, Peterson's solution, sleep & wake up (The Producer Consumer Problem), Semaphores

Unit 5: Scheduling
Basic concepts, primitive and non-primitive scheduling, types of scheduling - batch, interactive and real-time, scheduling algorithms, goals of scheduling algorithms, first come first serve, shortest job first and round robin scheduling.

Unit 6: Deadlocks
Definition of deadlock, principles of deadlock (deadlock conditions & modelling), deadlock detection, recovery & prevention, deadlock avoidance (banker‟s algorlithm)

Unit 7: Memory Management
Multiprogramming: fixed partitions, relocation and protection; Swapping and its basic concepts;Virtual Memory: Basic concepts, Paging and Page tables, Page Replacement Algorithms (FIFO,LRU; Cache Memory.

Unit 8: Memory Mapping
Associative Mapping, Set-Associative Mapping, Block Set-Associative Mapping, DMA.

Unit 9: File System
Definition of File, File naming, File types(directory, regular), Sequential access and Random access files, File attributes, Operations on file, Hierarchical directory structure, Path name(relative and absolute), Operation on directories, File system implementation techniques.

Unit 10: I/O Management
Basic principles I/O Operations, I/O Devices, Device controllers, DMA, Principles of I/O Software, its goals, Interrupt Handlers, Device Drivers.

Unit 11: Protection
Needs for protection, domain of protection, Example in UNIX, access matrix, Implementation of Access matrix

Unit 12: Security
The security problem, Authentication, Password, password vulnerabilities, Encrypted password,One time password(OTP), Biometrics and its types, Intrusion detection (definition only).

Unit 13: Multiprocessor Systems
Types of Multiprocessor Operating Systems, Multiprocessor OS Functions and Requirements, Multiprocessor synchronization

Unit 14: Distributed Operating Systems
Distributed Processing, Coping with Failures Models of Distributed systems, Remote procedurecalls, distributed Shared Memory, Distributed File Systems.

Unit 15: Introduction to Multiprogramming System
Queue management, I/O supervisors, memory management. File system, Disk and Drum Scheduling.

References:

1. Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson Education, Inc.
2. Tanenbaum, A. S., & Bos, H. (2021). Modern operating systems (4th ed.). Upper Saddle River, NJ: Pearson.

SEMESTER IV: Artificial Intelligence and Machine Learning

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 20, Practical: 30, Summative: 50)

Course objectives:

- To understand the fundamental concepts and principles of artificial intelligence (AI) and machine learning (ML).
- To gain practical experience with implementing AI and ML algorithms and models, and evaluating their performance.
- To explore current and emerging applications of AI and ML, and their potential impact on society and industry.

Course outcomes:

On completion of this course, the learners will be able to

- describe the fundamental concepts and techniques of AI and machine learning, including problem-solving using search algorithms, heuristic search, knowledge representation, and game playing.
- use machine learning techniques such as regression, classification, and clustering to analyze data and make predictions. They will also be able to apply different models such as k-nearest neighbors, support vector machines, and neural networks to solve real-world problems.
- use programming languages such as Prolog and Python to implement AI applications, including expert systems and natural language processing. They will also learn how to evaluate the performance of their models using metrics such as accuracy, precision, and recall.

Unit 1: Introduction
Definition of AI, AI problems, Underlying Assumption, AI techniques, AI areas

Unit 2: Problems and Search
Defining the Problem as a State Space Search, Production Systems, Problem characteristics, Production System Characteristics, Issues in the design of Search Programs, Additional problems

Unit 3: Heuristic Search Techniques
Generate-and-Test ,Hill Climbing ,Best-first Search ,Problem Reduction ,Constraint Satisfaction ,Means-end Analysis

Unit 4: Knowledge Representation

Representations and Mappings ,Approaches to Knowledge Representation ,Issues in Knowledge Representation ,Representing Simple Facts in Logic ,Resolution ,Natural Deduction ,Representing Knowledge using Rules ,Scripts, Frames, Semantic Nets

Unit 5: Game Playing
Overview, The Minimax Search Procedure, Adding Alpha-beta Cutoffs, Iterative Deepening

Unit 6: Expert Systems
Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition

Unit 7: Prolog – The Natural Language of AI
Introduction, Converting English to Prolog Facts and Rules, Goals, Prolog Terminology, Variables

Unit 8: Introduction to Machine Learning
Basics of Machine Learning, Necessity and Application

Unit 9: Data Science in Machine Learning
Data Preprocessing, Data Standardization and Normalization, Feature Reduction, Feature Selection,

Unit 10: Regression
Linear Regression, Gradient Descent and Stochastic Gradient, Newton Method, Probabilistic Interpretation of Linear Regression: Maximum Likelihood

Unit 11: Classification
K-Nearest Neighbours (KNN), Naive Bayes, Linear Discriminant Analysis/Gaussian Discriminant analysis Logistic Regression

Unit 12: Kernel Methods
Kernel Density Estimation, Kernel Regression, Support Vector Machines, Gaussian Processes

Unit 13: **Neural Networks**
Perceptron, MLP and Back Propagation, Deep Neural Network

Unit 14: Unsupervised Learning
Clustering: K-means, Gaussian Mixtures, PCA, Dimensionality Reduction

Unit 15: Application of AI/ML in Natural Language Processing
Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Statistical Natural Language Processing, Spell Checking

References:

1. Pradhan Manaswini, Fundamentals Of Artificial Intelligence And Machine Learning, Dps Publishing House

SEMESTER IV: Introduction to Cloud Computing

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative:30, Summative: 70)

Course objectives:

- To understand the fundamental concepts and characteristics of Cloud Computing, including the various deployment models, service models, and architecture components.
- To analyze the benefits and challenges of Cloud Computing and evaluate its suitability for different types of business and IT scenarios.
- To gain knowledge of the key technologies and tools used in Cloud Computing, such as virtualization, storage, security, and management platforms.
- To develop skills to plan, deploy, and manage Cloud Computing services and applications, including migration strategies, cost optimization, and compliance with standards and regulations.

Course outcomes:

On completion of this course, the learners will be able to

- design, implement, and manage Cloud Computing services and applications using different Cloud deployment models and service models.
- understand of the security, privacy, and compliance issues in Cloud Computing and ability to apply appropriate measures and controls to mitigate these risks.
- familiar with the emerging trends and technologies in Cloud Computing and ability to evaluate their potential impact on business and IT operations.

Unit 1: Introduction to Cloud Computing
Definition of Cloud Computing, Key characteristics of Cloud Computing, Advantages and disadvantages of Cloud Computing, Cloud deployment models

Unit 2: Cloud Service Models
Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Function as a Service (FaaS), Container as a Service (CaaS)

Unit 3: Cloud Computing Architecture
Components of Cloud Computing Architecture, Layers of Cloud Computing Architecture, Cloud Computing Service Providers, Cloud Computing Infrastructure

Unit 4: Virtualization and Cloud Computing
Virtualization technology, Types of Virtualization, Virtualization and Cloud Computing, Advantages and disadvantages of Virtualization in Cloud Computing

Unit 5: Cloud Storage and Database
Types of Cloud Storage, Advantages and disadvantages of Cloud Storage, Cloud Database and its characteristics, Advantages and disadvantages of Cloud Database

Unit 6: Cloud Security and Privacy
Security concerns in Cloud Computing, Cloud Security measures, Cloud Security models, Cloud Privacy concerns

Unit 7: Cloud Computing Platforms
Cloud Platforms and Services, Cloud Management Platforms, Cloud Development Platforms, Cloud Integration Platforms

Unit 8: Cloud Infrastructure Management
Cloud Infrastructure management tools, Cloud Automation, Cloud Monitoring, Cloud Resource Management

Unit 9: Cloud Applications and Services
Cloud Application development, Cloud Application services, Cloud Computing use cases

Unit 10: Cloud Migration and Deployment
Cloud migration challenges, Cloud migration strategies, Cloud Deployment models, Hybrid Cloud deployment

Unit 11: Cloud Computing Standards
Cloud Computing Standards and Frameworks, Cloud Computing Compliance and Regulations, Cloud Computing Governance

Unit 12: Cloud Computing Business Models and Costing
Cloud Computing Business Models, Cloud Computing Costing Models, Cost optimization strategies in Cloud Computing

Unit 13: Cloud Computing Case Studies
Cloud Computing success stories, Cloud Computing implementation case studies, Cloud Computing adoption trends

Unit 14: Future of Cloud Computing
Emerging trends in Cloud Computing, Advancements in Cloud Computing, Future of Cloud Computing in Business and IT industry.

References:

1. Erl, T. (2018). Cloud computing: Concepts, technology & architecture (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

SEMESTER IV: Programming in Python

Course Type:	DSC
Number of Credits:	4
Total Marks:	100 (Formative: 20, Practical: 30, Summative: 50)

Course objectives:

- Understand the basic syntax and data types of Python and be able to write and execute simple programs in Python.
- Develop an understanding of control structures and how to use them in Python programs to make decisions and control program flow.
- Be able to work with collections in Python, including lists, tuples, sets, and dictionaries, and understand their differences and use cases.
- Learn object-oriented programming concepts and how to use them in Python, including defining classes, objects, inheritance, and polymorphism, as well as exception handling, regular expressions, and file input/output.

Course outcomes:

On completion of this course, the learners will be able to:

- write and execute Python code that uses variables, data types, operators, control structures, functions, and basic input/output operations.
- familiar with the main data structures in Python, including lists, tuples, dictionaries, and sets. They will be able to create, modify, and access data stored in these structures.
- understand advanced topics in Python, such as object-oriented programming, regular expressions, file handling, and GUI programming. While they may not have mastered these topics, they will have a basic understanding of them and be able to continue learning and exploring them on their own.

Unit 1: Introduction to Python
Introduction to Python and its features, Installation of Python and setup of development environment, Running a simple program in Python

Unit 2: Variables and Data Types
Variables and their types in Python, Data types in Python, including integers, floats, strings, and Booleans; Working with variables and data types in Python

Unit 3: Operators and Expressions
Arithmetic, comparison, and logical operators in Python, Expression evaluation in Python, Order of precedence in Python expressions

Unit 4: Control Structures
Conditional statements in Python, including if-else and nested if-else statements; Looping statements in Python, including for and while loops; Control flow statements in Python, including break and continue statements

Unit 5: Functions
Introduction to functions in Python, Defining and calling functions in Python, Parameters and return values in Python functions

Unit 6: Lists
Introduction to lists in Python, Creating and manipulating lists in Python, Slicing and indexing in Python lists

Unit 7: Tuples
Introduction to tuples in Python, Creating and manipulating tuples in Python, Difference between lists and tuples in Python

Unit 8: Dictionaries
Introduction to dictionaries in Python, Creating and manipulating dictionaries in Python
Using dictionaries to solve problems

Unit 9: Sets
Introduction to sets in Python, Creating and manipulating sets in Python, Using sets to solve problems

Unit 10: File Handling
Introduction to file handling in Python, Reading and writing text files in Python, Handling errors while working with files

Unit 11: Object-Oriented Programming
Introduction to object-oriented programming in Python, Defining classes and objects in Python, Encapsulation, inheritance, and polymorphism in Python

Unit 12: Exception Handling
Introduction to exception handling in Python, Types of exceptions in Python, Handling exceptions in Python code

Unit 13: Regular Expressions
Introduction to regular expressions in Python, Defining regular expressions in Python, Using regular expressions to search and replace text in Python

Unit 14: Modules and Packages
Introduction to modules and packages in Python, Creating and importing modules in Python, Using third-party packages in Python

Unit 15: GUI Programming
Introduction to GUI programming in Python, Using Tkinter library for GUI programming in Python, Creating and running GUI programs in Python

Unit 16: Advanced Topics
Advanced topics in Python, including multi-threading, networking, and web programming, Using libraries and frameworks for advanced Python development, Best practices for Python

programming and development.

References:

1. Brown, M. C. (2020). Python: The complete reference (5th ed.). McGraw-Hill Education
2. https://www.w3schools.com/python/

M.SC.IT DETAILED COURSE WISE SYLLABUS

Syllabus of Discipline Specific Elective (DSE) Courses

SEMESTER I: PC Software

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formative:20, Practical: 30, Summative: 50)

Course objectives:

- To gain an understanding of the fundamental concepts and functions of operating systems, including the various types of operating systems and their applications.
- To develop practical skills in using MS DOS, Windows, and Linux operating systems, including file and directory management, command-line interface, and text editing.
- To learn the features and advantages of LibreOffice software suite, and develop proficiency in using its components, including Writer, Calc, Impress, and Base.
- To acquire practical skills in document formatting, spreadsheet calculations, presentation design, and database management using LibreOffice tools.

Course outcomes:

On completion of the course, the learners will be able to

- explain the fundamental concepts and functions of operating systems, and compare and contrast different types of OS.
- use MS-DOS, Windows, and Linux operating systems, including basic and advanced features such as file management, command line interface, and permissions.
- use LibreOffice suite to create, edit, format, and present documents, spreadsheets, and presentations, as well as to manage databases, using templates, styles, formulas, and advanced formatting features.

Unit 1: Introduction to OS

Definition and functions of an Operating System, Types of OS (Single User, Multi user, Single tasking, Multitasking, Real time, Network OS, Distributed OS)

Unit 2: MS DOS Operating System

Introduction to DOS, System files of DOS, concept of Booting, Files and Directory Structure, Concept of Paths, Internal and External commands, Batch File.

Unit 3: MS Windows Operating System

Introduction to Windows, Components of Windows, Customizing the Desktop, Files and Folders.

Unit 4: LINUX Operating System

Open Source and Free Software, Advantages and Disadvantages of Linux OperatingSystem, Concept of Path, Basic Linux Commands, File Permission, Text Editor vi.

Unit 5: Introduction to LibreOffice

Overview of LibreOffice software suite and its features, Advantages of using LibreOffice as an alternative to proprietary software, Installation and configuration of LibreOffice on different platforms

Unit 6: Getting started with Writer

Launching Writer and creating a new document, Navigating the Writer interface and understanding the different components of the document window, Entering and formatting text, applying font styles, and working with paragraphs, Saving, closing, and opening existing documents in different formats

Unit 7: Working with templates and styles in Writer

Creating and using document templates for consistent formatting and layout, Creating and modifying paragraph and character styles for consistent text formatting, Managing styles in the Styles and Formatting dialog box

Unit 8: Formatting documents in Writer

Adding and formatting images, tables, and other objects in a document, Applying advanced formatting options, such as page borders, columns, and page backgrounds, Using Writer's built-in tools for spell-checking, grammar-checking, and language translation

Unit 9: Introduction to Calc

Overview of Calc and its features as a spreadsheet program, Launching Calc and creating a new spreadsheet, Navigating the Calc interface and understanding the different components of the worksheet window, Entering and formatting data, applying cell formatting, and working with formulas

Unit 10: Working with functions and formulas in Calc

Using built-in functions and formulas for common calculations in Calc, Creating custom functions and formulas for complex calculations, Using conditional formatting to highlight data based on certain criteria

Unit 11: Advanced formatting and charting in Calc

Applying advanced formatting options, such as conditional formatting, data validation, and custom number formats, Creating and modifying charts to visualize data in different ways, Using Calc's built-in tools for filtering, sorting, and analyzing data

Unit 12: Introduction to Impress

Overview of Impress and its features as a presentation program, Launching Impress and creating a new presentation, Navigating the Impress interface and understanding the different components of the slide window, Adding and formatting slides, applying transitions, and customizing presentation settings

Unit 13: Working with multimedia and animations in Impress

Adding images, audio, and video to a presentation, Creating and modifying animations to enhance visual effects, Using Impress's built-in tools for recording and rehearsing presentations

Unit 14: Introduction to Base

Overview of Base and its features as a database management tool, Launching Base and creating a new database, Navigating the Base interface and understanding the different components of the database window, Creating tables, forms, queries, and reports using Base's built-in tools

References:

1. https://documentation.libreoffice.org/assets/Uploads/Documentation/en/GS7.0/GS70-GettingStarted.pdf

SEMESTER I: Open Source Software

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formative:20, Practical: 30, Summative: 50)

Course objectives:

- To familiarize learners with LaTeX and Scilab, two popular tools used in scientific and technical writing and computing, respectively.
- To enable learners to use LaTeX to create professional-looking documents that include equations, tables, figures, references, and a table of contents.
- To teach learners how to use Scilab for numerical computing tasks such as solving linear equations, calculating eigenvalues, and solving differential equations.
- To develop learners' programming skills in Python, including basic syntax, programming concepts, and applications.

Course outcomes:

On completion of the course, the learners will be able to

- use LaTeX to create professional-looking documents, including research papers, articles, and presentations, with advanced features such as equations, tables, figures, and indices.
- apply Scilab to perform numerical computations, including linear algebra, optimization, and differential equations, and visualize the results using 2D and 3D graphics.
- develop basic programming skills in Python, including understanding the syntax, using built-in functions, and creating simple programs for various applications.

Unit 1: Installation of LaTeX

Installation of LaTeX, Understanding Latex compilation

Unit 2: Introduction to LaTeX

Basic Syntax, Writing equations, Matrix, Tables

Unit 3: Page Layout – I

Page Layout – Titles, Abstract Chapters, Sections, References, Equation references, citation, Table of contents

Unit 4: Page Layout – II

List making environments, Generating new commands, Figure handling, table & figure numbering, List of figures, List of tables, Generating index

Unit 5: Packages

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

Unit 6: Classes

Classes: article, book, report, beamer, slides, letter

Unit 7: Applications of LaTex

Applications: Writing Resume, Writing question paper, Writing articles/ research papers,

$$(x+a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k},$$

$$(1+x)^n = 1 + \frac{nx}{1!} + \frac{n(n-1)x^2}{2!} + \cdots,$$

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\frac{n\pi x}{L} + b_n \sin\frac{n\pi x}{L} \right)$$

Presentation using beamer, inserting graphics, drawing graphics, putting equations like :

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots, -\infty < x < \infty$$

Unit 8: Installation of Scilab

Installation of Scilab (both windows & Linux)

Unit 9: Introduction to Scilab

Introduction to scilab, Basic syntax, Mathematical Operators, Predefined constants, Handling .scifiles, Installation of additional packages e.g. optimization

Unit 10: Functions in Scilab

Built in functions, Complex numbers, Polynomials, Vectors, Matrix operations (functions like inv(), spec(), zeros(), ones(), eye(), rand(). Handling these data structures using built in functions.Functions

Unit 11: Conditional Statements in Scilab

Loops (for & while), Conditional statements

Unit 12: Graphics in Scilab

Graphics handling: 2D, 3D, Generating .jpg files, Function plotting, Data plotting, GUI inscillab, Plotting 2D graphs

Unit 13: Applications of Scilab

Applications: Numerical Linear Algebra (Solving linear equations, eigen values atc.) solving Ordinary Differential Equations, Numerical Analysis – iterative methods, Comparison with C/C++/Matlab

Unit 14: Introduction to Python

Introduction to Python, The procedure to install Python, How to open Python console

Unit 15: Basic Python Commands

Basic Python commands, Python programming

References:

1. Smith, J. (2019). LaTeX for Beginners: An Introduction to the Typesetting System. APA Press.
2. Tao, Y. (2017). Scilab: A practical introduction to programming and problem solving. CRC Press.

SEMESTER II: Linux System Administration

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formative:20, Practical: 30, Summative: 50)

Course objectives:

- Understand the fundamentals of system administration, including the role of a system administrator and common administrative tasks performed in a Linux environment.
- Develop proficiency in installing and configuring a Linux operating system using different methods, such as direct installation, virtual machines, and live CDs.
- Gain knowledge of the Linux kernel, its device drivers, and how it interacts with hardware components to start and shut down the system.
- Acquire practical skills in managing files, directories, and the Linux file system, including how to perform basic file operations, mount and unmount file systems, and set file permissions and ownerships.

Course outcomes:

On completion of the course, the learners will be able to

- understand the fundamentals of system administration, including the role of a system administrator, common administrative tasks, and the identification of administrative files such as configuration and log files.

- demonstrate proficiency in installing, configuring, and using the Linux operating system, including knowledge of various Linux distributions, the Linux kernel and device drivers, and basic Linux file system management.
- develop skills in using Linux commands and filters, including file handling commands, filter commands, and shell commands, as well as an understanding of basic network protocols and network security issues.

Unit 1: Introduction to System Administration

Introduction to System Administration, Role of System Administrator, Qualities of good System Administrator, System Administration Common administrative tasks, identifying administrative files – configuration and log files

Unit 2: Introduction to LINUX Operating System

Basic idea on Proprietary, Open Source, Free Software etc, Basic Features of the Linux operatingsystem, Introduction of Various Linux Distribution (Red Hat Enterprise Linux, Cent OS, Fedora Projects, Debian Linux, Ubuntu, SUSE etc.)

Unit 3: Installation of LINUX Operating System

Installation Requirements, Partitioning the Hard drive in Linux, Installing the Linux system: Using Live CD, Virtual Machine, Direct Installation, Installing and Configuring software in Linux

Unit 4: LINUX Kernel

Linux kernel and device drivers, System Startup and Shutdown, Standard I/O, Standard error, Redirection and Piping

Unit 5: Basics of LINUX File System

Basics of Linux file system - File system types (ext3, ext4, xfs, jfs, ReiserFS, iso9660 etc.), Boot block, Super block, Inode table, Data blocks, How Linux access files, storage files, Linux standard directories, LILO, GRUB Boot Loader, three basic types of files (ordinary or regular, special or device and directory), I-nodes and file attributes, init and run levels

Unit 6: File organization

Files and File System (File Types and Permissions, Links, Size and Space, Date and Time); Working with Files: Reading Files, Searching for files, Copying, Moving, Renaming, Deleting, Linking, and Editing Files; Absolute and Relative path names, File system Mounting and

Unmounting, Organization of the file tree, Standard directories and their contents, Disk related commands, checking disk free spaces.

Unit 7: LINUX File Handling Commands

Files and Directory handling Commands - ls, cd, cp, mv, rm, mkdir, rmdir, pwd, Commands for Creating and Viewing ordinary files – cat, more, less, pg

Unit 8: LINUX Filter Commands

Filter Commands – wc, head, tail, cut, tr, top, grep (with regular expressions),. Other commands –tar, unzip, nice, kill, netstat. Setting user and group ownership of files and Access permissions –chmod, chown, chgrp commands

Unit 9: LINUX Shells

Various types of Shell available in Linux, Comparisons between various Shells, Study of different Linux Shells (sh, bash, csh, zsh), Environment variables, Shell script basics (examples of some simple shell programming), Shell programming in bash, read command, conditional and looping statements, case statements, parameter passing and arguments, Shell variables, system shell variables, shell keywords, creating Shell programs for performing various tasks.

Unit 10: Process

Services and Process, Basic commands for starting and stopping processes, Basic process attributes and their role in Access control. Examining the list of running processes on the system and understand the data presented there. Background process, Job control, Crontab file format, Backup and Restore procedure

Unit 11: Printing Jobs

Submit a print job, check the status of a print job, cancel a print job, configuring the Print Queue,Selecting the Print Driver, Editing the Printer configuration

Unit 12: System Administration Tasks

Getting Started: Login process, Creating Users Account and Group, Getting Help, Understandingthe "root" account, Managing user accounts-adding, modifying and removing User accounts, changing permissions and ownerships, Creating and managing groups, modifying group attributes, becoming super user using su; limited su (sudo) Changing Password, System monitoring and logging, Monitoring memory usage, disk space usage and I/O activity, Temporary disable user‟s accounts, creating and mounting file system, checking and monitoring system performance file security & Permissions, Getting system information with uname, host name, disk partitions & sizes, users, kernel. Backup and restore files, reconfiguration hardware with kudzu, installing and removing packages in Linux. Configure X-windows starting & usingX desktop. KDE & Gnome graphical interfaces, changing X windows settings.

Unit 13: Networking in LINUX

Installation and configuration of a simple LAN; Installation and configuration of: Proxy server(Squid), DNS server(BIND), Mail server, Web server(Apache), File server(Samba), DHCP server; Installation and configuration of a SSH server and client; Installation and configuration ofFTP server and client, rules governing IP address classes and netmasks, Network Address, Netmask and Gateway configuring Interface with ifconfig, ping, netstat, traceroute, telnet, understanding the significance of the /etc/services file and well known port numbers

Unit 14: Network Protocols

Basics of configuring NFS, NIS, DNS, FTP, Squid Proxy, DHCP server, ip tables and firewall

Unit 15: Basic Network Security Issues

Introduction to Basic Network Security Issues, Packet Snifflers, DOS Attacks, Linux Kernel Firewalling, Virtual Private Networks

References:

3. Nemeth, E., Snyder, G., Hein, T., & Whaley, B. (2017). UNIX and Linux system administration handbook. Addison-Wesley Professional.
4. Petersen, N., & Whillock, T. (2017). Linux administration: A beginner's guide, seventh edition. McGraw-Hill Education.

SEMESTER II: System Analysis and Design

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formatice:30, Summative: 70)

Course objectives:

- Understand the fundamental concepts of systems analysis and design, including the characteristics of systems, types of systems, and the role of a system analyst.
- Gain knowledge and skills in the various stages of the system development life cycle, from recognizing the need for a new system to implementing and maintaining it.
- Learn the tools and techniques used in structured analysis and design, such as data flow

diagrams, decision trees, and cost/benefit analysis.

- Develop proficiency in designing various system components, including input/output forms, databases, and testing plans, as well as project scheduling and management techniques.

Course outcomes:

On completion of the course, the learners will be able to

- explain the different types of systems and their characteristics, as well as identify the elements that make up a system such as inputs, outputs, processors, control, feedback, environment, boundaries, and interface.
- apply system development life cycle (SDLC) methodologies to identify business needs, analyze requirements, design and implement systems, and perform post-implementation maintenance.
- utilize structured analysis tools such as data flow diagrams, decision trees, and decision tables to model business processes and systems, as well as conduct feasibility studies and cost/benefit analyses to determine the viability of proposed systems.

Unit 1: System Concept

System definition, Characteristics of a system: Organization, Interaction, Interdependence, integration, Elements of a system: inputs and outputs, Processors, control, feedback, environment, boundaries and interface, Types of systems: Physical or abstract system, open or closed system.

Unit 2: System Development Life Cycle

Introduction, Recognition of need, Feasibility study, Analysis, design, implementation, post implementation and maintenance, consideration for candidate system, prototyping,

Unit 3: Role of System Analyst

Introduction, Interpersonal skill, academic and professional qualification, multifaceted role of theanalyst, analyst/user interface, conflict resolution, place of analyst in the MIS organization

Unit 4: System Planning and Initial Investigation

Introduction, bases for planning in system analysis, dimensions of planning, strategic MIS planning, Managerial and operational MIS planning, initial investigation, needs identification, determining the user‟s information requirements, background analysis, fact finding, review of written documents, on site observations

Unit 5: Information Gathering

Introduction, types of information needed, information about the organization, information aboutuser staff, information about work flow, origin of information, review of literature, procedures and forms, interviews and questionnaires

Unit 6: Tools of Structured Analysis

Introduction, Data flow diagram (DFD), DFD symbols, constructing a DFD, data dictionary, structured English, decision tree, decision table

Unit 7: Feasibility Study

Introduction, system performance definition, statements of constraints, identification of specific system objectives, feasibility consideration: economic feasibility, technical feasibility, behavioralfeasibility, steps in feasibility analysis, feasibility report

Unit 8: Cost/Benefit Analysis

Introduction, data analysis, cost and benefit categories, procedure for cost/benefit determination

Unit 9: The Process and Stages of System Design

Introduction, The process of design: Logical and physical design, design methodologies, structured design, functional decomposition, Forms driven methodology, HIPO and IPO chart

Unit 10: Input/Output and Form Design

Introduction, Input design, input data, source data, input media and devices, online data entry, formatted form, screen design, output design, form design, requirements of form design, types ofform, box design

Unit 11: Database Design

Introduction, objective of database, key terms in database, Logical and physical views of data, schemas and subschemas, Types of relationships, Types of data structure : hierarchical, network, relational, entity and attributes, role of dataset administrators, Nornalization

Unit 12: System Testing

Introduction, Need for system testing, nature of test data, test plan, activity network for system testing, types of system test

Unit 13: Quality Assurance

Introduction, quality factor specification, levels of quality assurance, software requirement specification, software design specification.

Unit 14: Implementation

Introduction, conversion, activity network for conversion, file conversion, creating test files, training aids

Unit 15: Project Scheduling

Introduction, reason for system failure, project management, Planning tools: Gantt charts, program evaluation and review techniques (PERT), project management software.

References:

1. Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2015). Systems analysis and design in a changing world. Boston, MA: Cengage Learning.
2. Dennis, A., Wixom, B. H., & Roth, R. M. (2015). Systems analysis and design. Hoboken, NJ: John Wiley & Sons.

SEMESTER III: Programming in Java

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formative:20, Practical: 30, Summative: 50)

Course objectives:

- To gain a comprehensive understanding of the Java programming language, including its basic features, syntax, and environment, and to be able to install the Java SDK and write Java programs.
- To develop proficiency in using Java operators and assignments, including arithmetic, comparison, logical, and bitwise operators, and to be able to cast variables and use conditional operators.
- To acquire the programming skills needed to develop Java applications, including knowledge of Java tokens and keywords, data types, variables, decision-making and control statements, and object-oriented programming concepts such as classes, inheritance, and interfaces.
- To learn advanced Java programming techniques such as exception handling, file handling, thread management, and database programming using Java, and to be able to apply these techniques to real-world programming problems.

Course outcomes:

On completion of the course, the learners will be able to

- write and execute simple Java programs using various features of the Java environment, including data types, variables, operators, and control statements.
- gain a solid foundation in object-oriented programming concepts, such as classes, objects, inheritance, polymorphism, and interfaces, and will be able to apply these concepts to design and implement Java programs.
- learn about advanced Java features, such as multithreading, exception handling, file handling, applets, AWT, Swings, and JDBC, and will be able to use these features to develop more sophisticated Java applications that interact with databases and other external resources.

Unit 1: Introduction to JAVA

An overview of JAVA, Basic features of Java, JAVA Environment, Installing the Java SDK, Writing Java Programs

Unit 2: Operators and Assignments

Introduction, Assignment operator, Comparison Operators, Instance of Comparison, Arithmetic operators, Shift operator, Bitwise operator, Logical operator, bitwise operator, conditional operator, casting.

Unit 3: Programming Basic

Java Token & Keywords, Constants, Data types; Declaring a variable, The scope and lifetime of variable, Various Operators, Input/Output statements Decision Making and Control Statements : if statement, If-else, else-if, switch statement; for, while, do-while statements

Unit 4: Class in Java

Class fundamentals: Defining class, Accessing class members, Declaring objects, Passing Arguments to Methods, Returning Multiple Values from methods, Modifiers

Unit 5: OOP in Java

Class fundamentals: Defining class, Accessing class members, Declaring objects, Constructors, copy constructor; Passing Arguments to Methods, modifiers, Inheritance: the super class, Multilevel Inheritance, Final and abstract keyword, Static Members

Unit 6: Arrays, Strings and Vectors

Declaring Arrays, Creating Arrays, Initializing Arrays, Multi-Dimensional Arrays, Strings: stringarrays, string methods, String Buffer class, Vectors

Unit 7: Interfaces and Packages

Interfaces: Defining an Interface, Implementing interfaces, Applying Interfaces, Packages: Defining a package, Accessing and Importing Packages

Unit 8: Threads

Introduction, new threads, creating new threads by extending the thread class, creating a thread by implementing Runnable Interface, Threads in the Running State, Sleeping and Interruptions, Signaling with wait, notify

Unit 9: The java.lang.Math class

Introduction, Methods of the java.lang.Maths: abs, ceil, floor, max, min, random, round, sin, cos,tan, sqrt, exp, log, pow.

Unit 10: Exception Handling

Exception Handling fundamentals, Exception types, Using *try* and *catch,* built-In exceptions in Java, User-defined exception

Unit 11: File Handling

I/O Basics: Streams, The Stream classes, The predefined streams, Reading console input, Writing console output, Reading and writing files

Unit 12: Introduction to Applets

Applets and the World Wide Web, The Applet Class, Applets and HTML , The Life Cycle of an Applet, Using Window Components, Event Handling, Adding Audio and Animation

Unit 13: AWT and Swings

AWT Basics, AWT Components, Event Handling, Application and Menus; Introduction to Swings, Swing Components, Event Handling, Display text and image in a window, Layout manager

UNIT 14: Introduction to JDBC

Basic steps to JDBC, API, JDBC Drivers, Connection Management, JDBC Design Considerations, Two Tier and Three Tier client server model, ResultSet, Prepared statement and callable statement, ResultSet Meta Data Object.

Unit 15: Database programming using Java

Keeping MySQL or NoSQL as backend and using java as front end good GUI should design to insert, update and delete record on a database table

References:

1. Horstmann, C. S. (2022). Core Java SE 15 for the Impatient. Addison-Wesley Professional.
2. Sierra, K., & Bates, B. (2021). Head First Java: A Brain-Friendly Guide. O'Reilly Media.

SEMESTER III: Computer Graphics

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formative:30, Summative: 70)

Course objectives:

- Understand the fundamental concepts and principles of computer graphics, including its applications in various fields such as CAD, CAM, entertainment, education, and image processing.
- Develop a strong foundation in graphics systems and software, including display technologies, graphics software, and algorithms for line drawing, circle and ellipse drawing, and filled area algorithms.
- Gain proficiency in 2D and 3D transformations, viewing, clipping, object representation, and projection, and learn how to apply these concepts to create computer animations and designs.

Course outcomes:

On completion of the course, the learners will be able to

- Develop a comprehensive understanding of the principles and techniques of computer graphics, including its applications in various fields such as CAD, CAM, entertainment, and education.
- Acquire proficiency in various line and shape drawing algorithms, filled area algorithms, transformations, and projections, and their implementation using graphics software such as GKS and PHIGS.
- Develop practical skills in 3-D object representation, animation designing, and polygon rendering methods, and gain hands-on experience with animation software such as Maya.

Unit 1: Introduction to Computer Graphics

Definition, Application Areas of Computer Graphics (CAD, CAM, Education and Training, Entertainment, Image Processing, Computer Art etc.), Graphical User Interfaces (GUI)

Unit 2: Graphics Systems

Cathode Ray Tubes, Random Scan Displays, Raster Scan Displays, Color CRT Monitors, Flat-Panel Displays (Plasma-Panels, Liquid Crystal Displays(LCD), Electroluminescent displays), Graphics Software (GKS, PHIGS)

Unit 3: Line Drawing Algorithms

Points and Lines, Line Drawing Algorithms (DDA Algorithm, Bresenham‟s Line Algorithm)

Unit 4: Circle and Ellipse Drawing Algorithms

Circle drawing algorithms, Ellipse Drawing algorithms

Unit 5: Filled area algorithms

Filling (Scan-Line Polygon filling, Inside outside tests, Boundary-fill and Flood-fill algorithm)

Unit 6: Transformations

Basic 2-D Transformations (Rotation, Reflection, shearing, scaling), Homogeneous CoordinateRepresentation, Translation, 3-D transformations

Unit 7: 2-D Viewing

2-D Viewing: The viewing pipeline, viewing coordinate reference frame, window to view-portcoordinate transformation, viewing functions

Unit 8: Clipping Algorithms

Line and polygon clipping algorithms (Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm)

Unit 9: 3-D Object representation

3-D Object representation: Polygon surfaces, quadric surfaces, spline representation, Basic illumination models, polygon rendering methods

Unit 10: Bezier and B-spline curves

Hermite curve, Bezier curve and B-spline curves, Bezier and B-spline surfaces

Unit 11: Projections

Projection Classification, Parallel projections, Perspective projections (One point, Two point),

Hidden Layer Projection

Unit 12: 3-D Geometric transformations

Translation, rotation, scaling, reflection and shear transformations, composite transformations

Unit 13: 3-D viewing

Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping

Unit 14: Basics in Animation

Animation Basic, Computer Animation, Principles of Animation, Types of Animation, Animation Software (Maya, etc) and Hardware, Common Terms in Animation

Unit 15: Animation Designing

Design of Animation sequence, Morphing, Application of Computer Animation, Future of Animation

References:

1. Shirley, P., Ashikhmin, M., & Marschner, S. (2018). Fundamentals of Computer Graphics. AK Peters/CRC Press.
2. Hearn, D., & Baker, M. P. (2017). Computer graphics with OpenGL (4th ed.). Pearson Education.

SEMESTER IV: Mobile Application Development

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formative:20, Practical: 30, Summative: 50)

Course objectives:

- To introduce students to the Android platform and its history, the Open Handset Alliance, and the architecture of the Android operating system.
- To provide students with the necessary knowledge and skills to install and use Android Studio for Android software development, including creating a new Android project, defining the project and SDK settings, and customizing the Android Studio IDE components.
- To enable students to design and develop Android applications, including understanding

the components of the Android manifest file, creating activities, services, and intents, and managing permissions and application resources.

- To teach students how to create user interfaces in Android applications, including working with basic user interface screen elements, designing user interfaces with layouts, and using drawing and animation features in Android. Additionally, students will learn how to work with data storage options, file I/O, preferences, SQLite databases, content providers, network data access, and web services in Android applications.

Course outcomes:

On completion of the course, the learners will be able to

- develop Android applications using Android Studio by gaining knowledge of the Android platform, Android versions, and factors that affect mobile application development.
- create efficient and user-friendly Android applications with proper user interface design, layout, and animation features.
- manage data storage options, file input/output, preferences, SQLite databases, and network data access in Android applications.

Unit 1: The Android Platform
History of mobile application development, The Open Handset Alliance (OHA), The Android Platform, Android Versions, Native Android Applications, Android SDK Features, Android Architecture, Factors that affect Mobile Application development

Unit 2: Installing Android Studio
System Requirements for Android Studio, Downloading Android Studio, Downloading JDK, Installing JDK, Installing Android Studio, Launching Android Studio

Unit 3: Android Studio for Android Software Development
Features of Android Studio, App Workflow, Android Virtual Devices (AVD), Using Hardware Device to test Application, Android Studio IDE Components, Android Studio Code Editor Customization.

Unit 4: Building a sample Android application using Android Studio
Building a sample Android application using Android Studio, Creating a new Android Project, Defining the Project and SDK Settings, Creating Activity,

Unit 5: Android Project Structure
Android Project Structure, Android Project Files, Android Application Modules, Types of Modules, Project structure settings, Anatomy of an Android Application, Important Android Terminology, Basic Android API Packages, Android Advanced API Packages

Unit 6: Android Manifest File and its common settings
Components of Manifest file, Package name and application ID, App components, Permissions, Device compatibility, File conventions, Manifest elements reference,

Unit 7: Android Application Design Essentials: Activities
What is activity?, Configuring the AndroidManifest.xml file, Life Cycle of an Activity, Understanding Life Cycle of an Activity, Context, Activity Transition

Unit 8: Android Application Design Essentials: Services
Use of services, Creating a service, Start and Stop Service, Service Life Cycle, Creating your own Service,

Unit 9: Android Application Design Essentials: Intents
Intent Structure, Other Operations on Intent, Types of Intent, Intent Resolution, Example of Intent, Explanation of Example, Standard Activity Actions, Standard Broadcast Actions

Unit 10: Android Application Design Essentials: Permissions
Permission approval, Request prompts for dangerous permissions, Permission for optional hardware features, Custom App Permission, Permission Protection Level, How to view App's permission?

Unit 11: Android Application Design Essentials: Application Resources
What are resources?, Resource Directory Hierarchy, Resource Value Types, Storing Different Resource Value Types, Accessing Resource Programmatically, Referencing System Resources

Unit 12: Basic User Interface Screen elements
Introduction to Views, Controls and Layout, TextView, EditText, AutoCompleteTextView, Spinner, Buttons, Check Boxes, Radio Groups, Pickers

Unit 13: Designing User Interfaces with Layouts
Creating Layouts Using XML Resources, Creating Layouts Programmatically, Built-In Layouts, Frame Layout, Linear Layout, Relative Layout, Table Layout, Data Driven Container

Unit 14: Drawing and Working with Animation
Canvas and Paints, Bitmaps, Shapes, Frame by Frame animation, Tweened Animation

References:

1. Deitel, P., Deitel, H., & Deitel, A. (2021). Android 11 for programmers: An app-driven approach with videos. Pearson.
2. Goyal, M. (2021). Android Application Development: Learn the Newest version of Android Operating System. Packt Publishing.

SEMESTER IV: Image Processing

Course Type:	DSE
Number of Credits:	4
Total Marks:	100 (Formative:20, Practical: 30, Summative: 50)

Course objectives:

- Develop a foundational understanding of digital image processing concepts and techniques, including image representation, sampling, transformation, enhancement, restoration, segmentation, compression, and binary and color image processing.
- Analyze the advantages and disadvantages of digital image processing compared to analog image processing, as well as the benefits and limitations of various image file formats and processing techniques.
- Apply digital image processing algorithms and techniques to real-world problems and scenarios, using software tools such as MATLAB, Python, or OpenCV, and evaluate the effectiveness of these solutions based on metrics such as image quality, computational complexity, and processing time.

Course outcomes:

On completion of the course, the learners will be able to

- develop a foundational understanding of digital image processing concepts and techniques, including image representation, sampling, transform, enhancement, restoration, segmentation, compression, and color processing.
- apply theoretical knowledge to real-world problems in image processing through practical exercises using popular software tools and programming languages.
- evaluate and compare different image processing techniques in terms of their advantages, limitations, and applicability to different types of images and applications.

Unit 1: Fundamental of image

Definition of image, Analog image, digital image, Advantages & disadvantages of digital image, digital image representation, pixel, neighbours of pixel

Unit 2: Image Sampling

Introduction, Theory of 2D sampling, violation of sampling criterion, quantization, Resolution, Anatomy of the human visual system

Unit 3: Digital Image

Classification of digital image : Raster image, vector image, Image types : Binary image, grayscale image, color image, multispectral image, hyper spectral image, Elements of image processing system : Image sensor, acquisition, CCD sensor, CMOS image sensor comparison of CCD and CMOS sensors, Digital camera, Image file format, GIF file format, JPEG, PNG, TIFF, PSD, EPS, BMP, Application of digital image processing

Unit 4: Image Transform

Introduction, need for transform, Image transform, classification of image transform ,Fourier transform, 2D discrete Fourier transform, Properties of 2D DFT: Separable, spatial shift property, periodicity, convolution, correlation properties, scaling property, rotation property

Unit 5: Image enhancement in spatial domain

Introduction, Image enhancement in spatial domain: point operation, mask operation, Types of point operation: Brightness modification, contrast adjustment

Unit 6: Image Histogram

Introduction, histogram equalization, procedure to perform histogram equalization

Unit 7: Filtering

Introduction, Spatial filtering, linear filtering, low pass filter, limitations of low pass filter, weighted average filter, Bartlett filter, Gaussian filter, Median filter, High pass filtering

Unit 8: Image enhancement in frequency domain

Introduction, Low pass filtering in frequency domain, High pass filter in frequency domain, Butterworth high pass filter, Homomorphic filter

Unit 9: Image Restoration

Introduction, Image degradation, Types of image Blur, Classification of Image Restoration technique, Image restoration model, Linear Image Restoration techniques, Pseudo Inverse filter, Wiener filter, Non Linear Image Restoration techniques : Iterative method, Maximum likelihood Method, Stochastic image restoration technique

Unit 10: Image noise

Introduction, classification of noise image: adaptive noise, Multiplicative noise, impulse noise, Median filtering, trimmed average filter

Unit 11: Image segmentation

Introduction, Classification of Image segmentation technique, Local and global segmentation, region approach to image segmentation, clustering technique, Image segmentation based on thresholding, Limitation, edge detection techniques, edge linking, Watershed transformation

Unit 12: Image compression

Introduction, Need for Image compression, Redundancy in images, Image compression scheme Huffman coding, Arithmetic coding, transform base coding, Vector quantization

Unit 13: Binary Image Processing

Introduction, Binarisation, Mathematical morphology, structuring element, Morphological image processing, Basic set theory, Logical operations, Standard binary morphological operations : Dialation, Erosion, Opening, Closing, Properties of Morphological operations

Unit 14: Color Image processing

Introduction, Color formation, Human perception of color, Color model: RGB model, CMY color model, HIS color model, YIQ color model, Histogram equalization of color image, Color Image filtering

Unit 15: Wavelet-based Image Processing

Introduction, Wavelet, Wavelet transform, 2D continuous wavelet transform, Discrete wavelet transform, Haar Wavelet, Filter bank, Embedded Image coding

References:

1. Jain, A. K., & Farrokhnia, F. (2021). Handbook of digital image processing: Concepts, algorithms, and scientific applications. CRC Press.
2. Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing using Python. Pearson Education.

ANNEXURE III

Project Guidelines of M.Sc.IT Programme

A. PROJECT GUIDE ELIGIBILITY CRITERIA:

Full Time Faculties in the Department of Computer Science/ Information Technology of KKHSOU/ Colleges/ Institutions affiliated to any Indian University recognized by UGC and having minimum 2 years teaching experience.

OR

A person having minimum M.Tech., MCA, M.Sc. in Computer Science/Information Technology from a UGC recognized universities with 4 years experience in Industry/teaching.

B. TYPE OF PROJECT

Learner may choose any topics according to M.Sc.IT standards. Most of the project work falls under the following types

a. Database oriented (e.g. payroll system, Loan management system etc.)

b. Application oriented (e.g. Mobile app development, web based application development etc.)

c. R & D project (e.g. Artificial Intelligence, Machine Learning, Image processing, speech processing, data mining, networking etc.)

C. PROJECT PROPOSAL (SYNOPSIS)

A project proposal or synopsis is a crucial framework that outlines the necessary details about a project. It provides a roadmap that helps project stakeholders understand the objectives, methodology, and expected outcomes of the project.

The necessary parts of a project proposal, as mentioned in the given format, help to structure the proposal and provide a comprehensive overview of the project.

i. Title of the Project: [Insert the title of your project here]

ii. Introduction and Objectives of the Project: [Provide a brief introduction to your project and describe the objectives you aim to achieve through it.]

iii. Project Category: [Specify the category of your project - RDBMS, application development, research and development, etc.]

iv. Tools, Platform, Hardware and Software Requirement Specifications: [List the tools, platforms, hardware, and software required for carrying out your project.]

v. Whether the Project is Done for Any Industry/Client? The Name and Address of the Industry or Client is to be Mentioned: [Specify whether your project is being developed for a particular industry or client. If yes, mention the name and address of the industry or client.]

vi. Methodology and Expected Output: [Provide details of the methodology you will be using to carry out your project and describe the expected output.]

vii. Conclusion: [Summarize your project proposal and highlight its significance.]

Consultation with a project guide is crucial to ensure that the project proposal/synopsis meets the required standards and expectations. A project guide or supervisor has significant experience and knowledge in the respective field, and their feedback can help refine the proposal and identify any gaps or potential issues that need to be addressed before the project begins.

Moreover, project guides can provide valuable insights and feedback that can enhance the quality of the project and improve the chances of its success. Therefore, it's essential to involve your project guide from the outset and maintain regular communication throughout the project's lifecycle to ensure that it's on track and aligned with the overall objectives.

D. APPLICATION AREAS & RELATED TOOLS

The following is a list of application areas and related tools that can be considered for developing a project. The chosen area will depend on the project's objectives, scope, and requirements.

Application Areas:

- Financial
- Manufacturing
- Multimedia
- Computer Graphics
- Instructional Design
- Database Management System
- Internet/Intranet
- Computer Networking-Communication Software
- E-Commerce
- TCP/IP Internals
- Routing Protocols
- Implementation of Switches & Routers

- Image Processing
- Mobile Apps Development

Related Tools:

- Front End/GUI Tools: PHP, scripting languages, etc.
- RDBMS/Back End: Oracle, MySQL, NoSQL, MongoDB, etc.
- Languages: C, C++, Java, VC++, C#, Matlab, Python, Scilab, etc.
- Internet Technologies: DHTML, JavaScript, VB Script, HTML, Java, Active X, SWING, JSP, ASP, PHP, XML, Java Beans, Java Servlets, CSS, VB.Net, AWT, J2EE.
- Networking Technologies: ATM, Frame Relay, TCP/IP, SNMP, GSM, VoIP, PPP, IP-PSTN, SONET/SDH
- Wireless Technologies: Bluetooth, 3G, ISDN, EDGE
- Operating Systems: Windows, DOS, UNIX, Linux, Android.
- This comprehensive list covers a wide range of areas and tools that offer diverse opportunities for project development. However, the selection of a specific area and related tools will depend on the project's objectives and requirements.

E. PROJECT REPORT GUIDELINE

The Project report should be prepared in well-structured preferably typed in Latex. Depending on the type of project the report should be as follows:

(a) For Database project:

i. Declaration: This page should include a statement declaring that the work presented in the report is the student's original work.

ii. Certificate: This page should include a certificate from the institution confirming that the student has completed the project.

iii. Certificate from Guide: This page should include a certificate from the project guide confirming that the student has completed the project under their supervision.

CHAPTER I: INTRODUCTION

1.1 Brief idea about the project

1.2 Objective of the project

1.3 Scope of the project

1.4 Existing system

1.5 Proposed System

(b) **R & D/ Application project**:

- Acknowledgement
- Content with page number
- Declaration Certificate
- Certificate from Guide

CHAPTER I: INTRODUCTION

1.1 Brief idea about the project
1.2 Objective of the project
1.3 Scope of the project
1.4 Application of the project
1.5 Proposed System
1.6 Platform used(Hardware & Software)
1.7 Project location

CHAPTER II: LITERATURE REVIEW

2.1 Introduction
2.2 Work already done in the area(Historical evidence)
2.3 Problem in Existing technology
2.4 Conclusion

CHAPTER III: THEORITICAL BACKGROUND

3.1 Introduction
3.2 Theory used in the project
3.3 Conclusion

CHAPTER IV: RESULT AND DISCUSSION

4.1 Introduction
4.2 Methodology
4.3 Result
4.4 Analysis on result
4.5 Conclusion

CHAPTER VI: Conclusion & Future work

6.1 Introduction
6.2 Chapter wise conclusion
6.3 Future work

References Appendix(if any)

CERTIFICATE OF ORIGINALITY FROM THE GUIDE

This is to certify that the project report entitled ....submitted to Krishna Kanta Handiqui State Open University in partial fulfilment of the requirement for the award of the degree of MASTER OF SCIENCE IN INFORMATION TECHOLOGY(M.Sc.IT), is an original work carried out by Mr./Ms.. Enrolment No. …………………under the supervision of Dr./Mr./Ms..The matter embodied in this project is a genuine work done by the student and has not been submitted either to this University or to any other University/Institute for the fulfilment of the requirement of any course of study.

Signature of the Learner

Name
Address
Enrolment No.:

Signature of the Guide

Name
Designation
Address

Study
Centre's
Seal

FORMAT OF THE PROJECT REPORT

KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

A Project Report on

Font type: Arial
Size: 12, Bold

TITLE OF THE PROJECT

Font type: Arial
Size: 16, Bold

In fulfillment of the requirement for the 4th Semester of
Master of Science in Information Technology Programme

Submitted by
....
(Name of the Learner)
Enrollment No.: ....
Session: ....

Font type:
Arial

Under the Guidance of
....
(Name of the Project Guide)

Study Centre
....
(Name of the Study Centre)
....
(Location)