SECOND YEAR (*Third Semester*)

Digital logic Design

Subject Code:		**Total Contact Hour**	**30**
Semester:	3rd	**Total Credit**	**03**
Subject Name:	**Digital Logic Design**		
Pre-requisites (if any):			
Course Objectives	1. Provide students with basic idea of digital logic concepts. 2. Familiarize students with basic digital logic gates and their operations. 3. Expertise students with combinational logic circuits. 4. Analyze the operations of sequential logic circuits and designs. 5. Perform design of different Sequential Circuits.		
SYLLABUS			
Module I	Introduction to Digital Systems: Introduction to Digital electronics, Digital and Analog Signals and Systems, Binary Digits, Logic Levels, Logic SystemsPositive and negative, Combinational and Sequential Logic Functions, Fixed-Function Logic Devices, Programmable Logic Devices. Number Systems and Codes: Introduction to Number Systems (Decimal, Binary, Octal and Hexadecimal), Conversion from one number system to other, Binary arithmetic operations, Representation of Signed Binary Numbers, BCD code, EBCIDIC code, Excess -3 code, Gray code, ASCII code.		**04 Hours**
Module II	Logic Gates: Logic variables, Logic Operators, Basic Logic Gates, Universal Gates and realization of Basic Gates using Universal Gates, Gate Delay. Boolean Algebra: Rules and laws of Boolean algebra, Demorgan's Theorems, Boolean expressions and Truth Tables, SOP and POS forms, Minterm and Maxterms, Canonical forms of Boolean expressions, Standard forms, Minimization Techniques for simplification of Boolean expressions using Karnaugh Map, Design of simple Logical Circuits.		**08 Hours**
Module III	Combinational Circuits: Introduction to combinational Circuits, Half Adder and Full Adder, Half Subtractor and Full Subtractor, Ripple/Parallel adder, Adder-Subtractor, Look-Ahead Carry Adder. BCD Adder, BCD Subtractor, BCD to Excess-3 and Excess-3 to BCD Code Converter, Binary to Gray and Gray to Binary Code Converter, Comparator, Parity Generator and Checker. Multiplexer, De multiplexer, Decoder, BCD to Seven Segment Display Decoder, Encoder, Priority Encoder.		**08 Hours**

Module IV	Sequential Circuits: Introduction to Sequential Circuits, Flip Flops-SR, D, JK, T, Triggering of Flip Flops, Master-Salve JK, Sequential Circuit Design. Shift Registers: Introduction to shift registers, Basic Shift Register Operations, types of shift registers, Bidirectional Shift Registers, Universal Shift Register.	**06 Hours**
Module V	Counters: Introduction to counters, Synchronous and AsynchronousCounters, Decade Counter, Ripple Counter, Up/Down Counter, Ring Counter and Twisted Ring Counter.	**04 Hours**
Essential Readings	1. M. Morris Mano, Michael D. Ciletti, "Digital Design", Prentice Hall of India Pvt. Ltd. / Pearson Education Pvt. Ltd. 2. Donald P. Leach and Albert P.Malvino, "Digital Principles and Applications", TMH.	
Supplementary Readings	1. John F.Wakerly, "Digital Design", Pearson/PHI. 2. Donald D. Givone, "Digital Principles and Design", TMH. 3. Charles H.Roth. "Fundamentals of Logic Design", Thomson Learning. 4. G. K. Kharate, "Digital Electronics", OXFORD University Press. 5. Thomas L. Floyd, "Digital Fundamentals", Pearson Education Inc.	
Course Outcomes	CO1: Define and memorize concepts of digital circuit operation and principles, CO2: Conceptualize and discuss different types of logic circuits, laws and theorems. CO3: Apply the knowledge of Boolean algebra, rules, theorems and concepts to design and demonstrate various digital circuits. CO4: Analyze, compare and differentiate the operations of various sequential logic circuits. CO5: Design various sequential circuits.	

Data Structures

Subject Code:		**Total Contact Hour**	**30**
Semester:	**3rd**	**Total Credit**	**3**
Subject Name:	**Data Structures**		
Pre-requisites (if any):			
Course Objectives:	1. Introduce the basic idea of data structure, arrays, linked lists, stacks, queues, and algorithms. 2. Explore the linear data structures linked lists, Stacks, and Queues in more detail. 3. Elaborate on non-linear data structures such as Graphs,Trees, BST, Spanning trees, etc. 4. Discuss the Sorting and Searching algorithms and their operations. 5.Study the different hashing techniques in detail.		

SYLLABUS		
Module I	**Introduction:** Introduction to Data structures and Algorithms, Analysis of Algorithms,Asymptotic notations, Time and space trade-off, Abstract Data Type.Arrays, Row/Column major representation of Arrays, Sparse matrix.	**4**
Module II	**Linked lists:** Definition, types of linked list (Single, Double, Circular), operations on linked list, Application of linked list **Stack:** Representation, operations, and applications of Stack. **Queue:**Representation, operations, and applications. Types of Queues (Circular, Priority, Deque).	**7**
Module III	**Tree:**Introduction to tree, Binary tree - definitions and properties; binary tree traversal algorithms with and without recursion. Binary Search Tree (BST):Operations on BST, AVL tree balancing; B-tree; B+ tree, Heap. **Graph:** Representation, Traversals (BFS and DFS).	**8**
Module IV	**Sorting and Searching:Sorting:** Internal **vs.** External sorting, Bubble,Insertion, Selection, Merge sort,Quick sort, Heap sort, Radix,**Searching:** Linear, Binary Search.	**6**
Module V	**Hashing:**Introduction, Hashing techniques, Hash function, Address calculation techniques- common hashing functions. Collision resolution techniques, Linear probing, quadratic probing, Double hashing and Rehashing.	**5**
Essential Reading	1.Introduction to Data Structures with Applications by J. Tremblay and P. G. Sorenson (TMH) 2.Classic Data Structures – Debasis Samanta (PHI)	
Supplementary Reading	1.Data Structures Using C – A.M. Tenenbaum (PHI) 2. Data structures with C-by Seymour Lipschutz (Schaum Outline Series) 3.Data Structures and Algorithm Analysis in C – M. A. Weiss (Pearson Education) 4. Fundamentals of Data Structures in C -- by Horowitz, Sahni, and Anderson-Freed (Silicon Press 2007). 5. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A. Forouzan, Cengage Learning.	
Course Outcomes:	After completion of the course successfully, students will have: **CO1**: Ability to understand the data structure and its application. **CO2:** Proficiency in selecting an efficient linear data structure and apply to solve its problem. **CO3:**Expertise in assessing efficiency trade-offs among different non-Linear data structures and implementations. **CO4:**Ability to apply Sorting and Searching operations in real-world problem solutions. **CO5:**Ability to design the programs using different data structures and hashing approaches.	

Database Engineering

Subject Code:		**Total Contact Hour**	30
Semester:	3rd	**Total Credit**	**3**
Subject Name:	**DATABASE ENGINEERING**		
Pre-requisites (if any):			
Course Objectives:	1. To know the basic concepts and architecture associated with DBMS and understand the data models. 2. To learn the various operations of relational algebra and know how to create, maintain, and manipulate a relational database using SQL commands. 3. To understand the fundamentals of relational database design and apply normalization steps on the database design for removal of data anomalies 4. To learn the query processing steps and select the most efficient strategy by applying query optimization algorithm. 5. To understand database transaction processing, concurrency control schemes and recovery from database failure.		
SYLLABUS			
Module I	**Introduction to Database and Data Models:** Basics concepts of Databases and Database Management Systems, Database System Architecture: Data Abstraction, Data Independence, Database Schema, Three-Schema Architecture, Data models: Entity-relationship model and Relational model.		**6**
Module II	**Relational Query Languages:** Relational Algebra, Relational Calculus, Introduction to Structured Query Language (SQL), Data Definition Language (DDL), Data Query Language (DQL), Data Manipulation Language (DML), Data Control Language (DCL), Integrity Constraints.		**7**
Module III	**Relational Database Design:** Introduction to Relational database design, Functional dependencies, Armstrong's Axioms, Dependency Preservation, Lossless design, Introduction to Normalization, Normal Forms: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF.		**8**
Module IV	**Query Processing and Optimization:**Basic Steps in Processing an SQL Query, Catalog Information for Cost Estimation, Measures of Query Cost, Selection Operation, Join Operations, Equivalence Rules, Query Optimization.		**4**
Module V	**Transaction Processing:**Transaction Concepts, Desirable Properties of Transactions, Schedules, Serializability of Schedules. **Concurrency Control and Recovery:**Concurrency Control, Concurrency Control Schemes: Lock-		**5**

	BasedSchemes, Time Stamping Methods, Database Recovery.	
Essential Reading	1. Elmasri & Navathe, "Fundamentals of Database systems", Pearson Education. 2. A. Silberschatz, H. F. Korth, S. Sudarshan, "Database System Concepts", McGraw Hill International Edition.	
Supplementary Reading	1. BipinC. Desai, "An Introduction to Database Systems", Galgotia Publication. 2. C. J Date, "An Introduction to Database Systems," Pearson Education India. 3. Ramakrishnan R &Gehrke, Database Management Systems (McGraw-Hill)	
Course Outcomes:	CO1: Understand the basic concepts, database modeling and architecture associated with DBMS. CO2: Use DDL and DML to query, update, and manage a database CO3: Understand the need of normalization and the various normal forms for a good relational database design CO4: Gain knowledge on the basics of query evaluation techniques and query optimization. CO5:Understand the basic issues and concepts associated with transaction processing and concurrency control.	

Object-Oriented Programming

Subject Code:		**Total Contact Hour**	**30**
Semester:	3rd	**Total Credit**	**3**
Subject Name:	**Object Oriented Programming**		
Pre-requisites (if any):	**C Programming**		
Course Objectives:	**1.** To understand principles of object-oriented programming in a higher-level programming language. **2.** Analyze a problem statement and develop program using class, object and basic concept of object-oriented programming. **3.** Utilize polymorphism and object-oriented concept to frame object-oriented programming. **4.** Gain skills in designing, and programming for reuse of code using inheritance **5.** Establish development methods in object-oriented programming for exception handling and templates.		
SYLLABUS			**Contact Hours**
Module I	Principles of object-oriented programming: Object oriented programming (OOP) paradigm, basic concepts, Benefits of OOP, Disadvantages of conventional programming, Beginning with C++. Evolution of C++, Application of OOP. **Classes and Objects:**Basic structure of OOP. Declaring classes and objects. Class Access-specifier: public, private and protected. Defining member functions,Characteristics of member functions,classes, objects and memory, array of objects,local classes,		**5**

	new and delete operator.	
Module II	**Functions In C++:** The mainfunction,function prototype, call by reference, return by reference, returning morevalues by reference, default arguments, constant argument, inline functions, Rules for inline functions. Static variable, function and object. Friend function. Recursive member function. Object as arguments **Constructors And Destructors**: Introduction, Characteristicsof constructors and destructors, types of constructors, overloading constructors, constructors with default arguments. Copy constructors, Dynamic constructor. Dynamic initialization using constructors.Destructors. Calling constructors and destructors, Anonymous objects.	**7**
Module III	**Polymorphism**: Introduction to polymorphism and types. Functionoverloading, operator overloading, overloading of unaryoperators,overloading of binary operators, Overloading with friend function. Rules for overloadingoperators. Type conversion.	**5**
Module IV	**Inheritance:**Introduction, types of inheritance: single inheritance,multilevel inheritance, multiple inheritance, hierarchical inheritance, hybrid inheritance,multipath inheritance, pointer to derived classes and base classes, pointer tomembers, Accessing private members with pointers. Constructors, Destructors and polymorphism in inheritance, Object asa class member, Virtual class, function and pure virtual function. Abstract Classes.	**7**
Module V	**Exception handling**: Introduction, Principles of Exception handling, the keyword try, throw and catch, Exception handling mechanism, multiple catch statements, Catch multipleExceptions, Re-throwing Exception. **Generic programming with Templates**: Introduction, need of template, Definition of class template, Normal function template, working of function templates, Class template with more parameters, Function template with Arguments and multiple parameters.	**6**
Essential Reading	1. E. Balagurusamy – Object Oriented Programming with C++, TMH publication. 2. Ashok N. Kamthane- Object oriented programming with ANSI & Turbo C++,PearsonEducation.	
Supplementary Reading	1. Programming with C++, ReemaThareja, Oxford University Press 2. H. Schildt – C++, The Complete Reference, TMH	
Course Outcomes:	CO1: Implement basics of object-oriented programming CO2: Apply object-oriented concept to implement programs using classes, objects. CO3: Demonstrate polymorphic behavior of objects and apply in real-life problem statement. CO4: Analyze and implement programs using Inheritance. CO5: Apply object-oriented approach to develop software incorporated with exception handling and templates.	

Digital Logic Design Lab

Subject Code:		**Total Contact Hour**	**20**
Semester:	3rd	**Total Credit**	**1.5**
Lab Name:	**Digital Logic Design Lab**		
Course Objectives:	1. Provide students basic idea of realization of Logic Gates. 2. Familiarize students with Boolean algebra and different circuit simplification techniques. 3. Analyze the operations of combinational logic circuits functions. 4. Expertise the students for implementation of logic circuits with minimum Gates. 5. Test and verify the implementation knowledge of students by a small project.		

Sl. No.	**Expt. No.**	**ExperimentDetails**
1	(i)	Realization of Logic Gates.
	(ii)	Realization of basic Logic Gates using Universal Logic Gates.
2		Design Logic Circuit of some given Boolean expressions after simplification using Boolean Algebra.
3		Design Logic Circuit of some given Boolean expressions after simplification using K-Map.
4	(i)	Design Half and Full Adder.
	(ii)	Design Half and Full Subtractor.
5	(i)	Design 3-bit Adder-Subtractor circuit using Logic Gates.
	(ii)	Design 3-bit Adder-Subtractor circuit using Adders.
6		Design 4-bit BCD Adder.
7	(i)	Design BCD to Excess-3 code convertor.
	(ii)	Design Excess-3 to BCD code convertor.
8	(i)	Design Binary to Gray code converter.
	(ii)	Design Gray to Binary code converter.
9		Design 2-bit/4-bit Comparator.
10		Small Hardware Project design.

Course Outcomes:	CO1: Realize the concepts of digital Logic Gate operations and principles, CO2: Conceptualize different types of Boolean Algebra laws, rules and theorems for circuit simplification. CO3: Apply the knowledge of Boolean algebra, rules, theorems and concepts to design and demonstrate various digital circuits with minimum Logic Gates. CO4: Analyze, compare and differentiate the operations of various combinational circuits. CO5: Design, implement and test various logic circuits.

Data structures lab

Subject Code:			**Total Contact Hour**	**20**
Semester:		**3rd**	**Total Credit**	**1.5**
Lab Name:		**Data structures lab**		
Course Objectives:		1. To implement data structures and analyze them for real-world problem-solving. 2. To implement and analyze the searching algorithms in the context of specific engineering problems. 3. Study to choose the appropriate data structure and algorithm design method for a specified application. 4. Understand the different data structures to apply in different application problem scenarios.		
Sl. No.	**Expt. No.**	**Experiment Details**		
1		Write a C Program for Traversal, Insertion, and Deletion operations of elements in an array.		
2		Write a C Program to create a stack and perform stack operations (using array).		
3		Write a C Program to create a queue and perform Queue operations.(using array)		
4	(i)	Write a C Program that uses Stack Operations to perform conversion of an infix expression into a postfix expression.		
	(ii)	Write a C Program that uses Stack Operations to perform evaluating of the postfix expression		
5	(i)	Write a C Program that uses functions to perform the following operations on a single linked list: i) Creation, ii) Traversal, iii) Insertion, iv) Deletion.		
	(ii)	Write a C Program that uses functions to perform the following operations on a double-linked list: i) Creation, ii) Insertion, iii) Deletion.		
6		Stack and Queue Operations using Linked List.		
7		Write a C Program that uses functions to implement the Binary Tree algorithm to perform the following operations: i) Traversal, ii) Creation, iii) Insertion, iv) Deletion.		
8		Write C Programs to implement code on: i) Bubble sort, ii) Selection sort, iii) Insertion sort, iv) Heap sort		
9	(i)	Write C Programs to implement code on: i) Quick sort, ii) Merge Sort		
	(ii)	Write C Programs to demonstrate operations on: i) Sequential Search, ii) Binary Search.		
10		Write a c program on the graph implementation and its DFS and BFS		

		methods.
Course Outcomes:		**CO1:** Ability to learn and implement operations performed on data structures. **CO2:** Improved skill in choosing and developing data structure applications in real-world scenarios. **CO3:** Potential to demonstrate and implement different linked list operations. **CO4:** Ability to develop different tree structures such as a binary tree, BST (Binary Search Tree), Heap tree, etc. **CO5:** Expert in choosing and developing searching and sorting algorithms suit for given scenarios.

DATABASE ENGINEERING LAB

Subject Code:			**Total Contact Hour**	**20**
Semester:		3rd	**Total Credit**	**1.5**
Lab Name:		**Database Engineering Lab**		
Course Objectives:		1) To know the fundamentals of MySQL and be familiar with SQL syntax of various operations. 2) To know how to create, maintain, and manipulate a relational database using SQL commands. 3) To know how to combine rows from two or more *tables* based on a related column between them. 4) To be familiar with the usage of arithmetic operators, conditional restrictions, logical operators and SQL aggregate functions. **5)** To acquire knowledge on writing sub-queries and views.		
Sl. No.	**Expt. No.**	**Experiment Details**		
1		Introduction to MySQL and basic commands for creating a database using CREATE DATABASE command and viewing it by SHOW DATABASES command.		
2	(i)	Build a database by creating table structures using the various data types and the CREATE TABLE command in MySQL.		
	(ii)	Apply various SQL constraints (NOT NULL, UNIQUE, PRIMARY KEY, DEFAULT, etc.) to the MySQL tables.		
3	(i)	Use of MySQL for deleting tables and displaying the structure of an individual table,		
	(ii)	Use of MySQL to list all the tables that have been created within a database, altering a table structure by ALTER TABLE command.		
4	(i)	Use of MySQL to INSERT, UPDATE and DELETE data from within a table.		
	(ii)	Use of MySQL to retrieve data from a table using the SELECT statement.		
5		Use of MySQL to apply arithmetic operators in SQL statements.		
6		Use of MySQL to select rows from a table with conditional restrictions.		
7	(i)	Use of MySQL to apply logical operators to combine multiple		

		conditions.
	(ii)	Write queries in MySQL on given exercises involving arithmetic operators, conditional restrictions and logical operators.
8	(i)	Use of MySQL to sort the data in the resulting query by applying ORDER BY clause (ascending (**ASC**) or descending (**DESC**)).
	(ii)	Use of MySQL to perform mathematical summaries through the use of aggregate (or group) functions.
9		Use of MySQL to perform join operations that merges rows from two or more tables satisfying certain join condition.
10	(i)	Use of MySQL to create sub-queries in MySQL.
	(ii)	Write query to create views in MySQL using CREATE VIEW command.
Course Outcomes:		CO1: Be familiar with fundamentals of MySQL and SQL syntax of various operations. CO2: Be able to create, maintain, and manipulate a relational database by applying appropriate SQL commands. CO3: Be able to combine rows from two or more *tables using* different join operations. CO4: Be familiar with the usage of usage of arithmetic operators, conditional restrictions, logical operators and SQL aggregate functions. CO5: Be able to create SQL sub-queries and views.

Object Oriented Programming Lab

Subject Code:			**Total Contact Hour**	**20**
Semester:		3rd	**Total Credit**	**1.5**
Lab Name:		**Object Oriented Programming Lab**		
Course Objectives:		**1.** To understand principles of object-oriented programming in a higher-level programming language. **2.** Analyze a problem statement and develop program using class, object and basic concept of object-oriented programming. **3.** Utilize polymorphism and object-oriented concept to frame object-oriented programming. **4.** Gain skills in designing, and programming for reuse of code using inheritance **5.** Establish development methods in object-oriented programming for exception handling and templates.		
Sl. No.	**Expt. No.**	**Experiment Details**		
1	(i)	Study of C++ Standard library functions in object oriented programming (OOP).		
	(ii)	Write a program illustrating class declarations, definition, and accessing		

		class members.
2	(i)	Write a program on static variable, static function and static object.
	(ii)	Program using inline functions in object oriented programming
3	(i)	Program on function overloading.
	(ii)	Write a program to demonstrate friend function and friend class.
4		Write a program to demonstrate the use of Constructor and Destructor in OOP.
5		Write a program to illustrate unary and binary operator overloading.
6		Write a program on type conversion
7		Write a program to demonstrate types of inheritance.
8		Write a program on function overriding.
9	(i)	Write a program using exception handling
	(ii)	Write a program to demonstrate the catching of all exceptions.
10		Write a program using function template and class template
Course Outcomes:		CO1: Define and memorize basics implementation of of object-oriented programming(OOP) CO2: Design and implement class and object to solve problem CO3: Demonstrate polymorphic behavior of objects and apply in real-life problem statement. CO4: Analyze and implement programs using Inheritance. CO5: Able to establish development methods in object-oriented programming for exception handling and templates.

SECOND YEAR (*Fourth Semester*)

Computer Organization and Architecture

Subject Code:		**Total Contact Hour**	**30**
Semester:	4th	**Total Credit**	**03**
Subject Name:	**Computer Organization and Architecture**		
Pre-requisites (if any):	**Concept of Digital Logic Design**		
Course Objectives	1. Provide students with basic idea of Different components of the Computer System and Computer arithmetic. 2. Familiarize students with the Instruction set Architecture and CPU organization. 3. Expertise students with Memory Design and Memory Operations, Memory characteristics. 4. Analyze the I/O operations, Compare different data transfer techniques and modes of data transfer. 5. Perform performance analysis of the system.		
SYLLABUS			
Module I	**Introduction:** Basic Organization of Computers, Basic Operational concepts, Registers, Data bus, Address bus, Control bus, Types of Bus, Concept of Harvard Architecture and Von-Neumann Architecture, IAS Computer. **Computer Arithmetic:** Binary Arithmetic operation,Decimal Arithmetic Operation, Floating Point Representation and Arithmetic operation, General Multiplication, Booth Multiplication and Division Algorithms, Array Multipliers.		**06 Hours**
Module II	**Instruction Set Architecture:** GeneralInstruction Format, Three Address, Two Address, One Address and Zero Address Instruction, Addressing Modes, Types of Instruction, Instruction Cycle. **CPU Organization:** Data Path**,** Singlebus Data Path**,** Register transfers, Fetching and storing a word in Memory, Control sequences for operation of an Instruction, Multi bus Data Path, Simple ALU Design, Control Unit Operation: Hardwired Control Unit and Micro Programmed Control Unit, Control Word, Stack Organization, RPN, Evaluation of Arithmetic expression using RPN, Subroutine, Nested Subroutine.		**08 Hours**
Module III	**Memory Organization:** Computers Memory System Overview, Characteristics of Memory System, Memory Hierarchy, Memory Classification, Semi Conductor Memory Organization, Memory Cell Operation. Cache Memory: Cache Principles, Levels of Cache, Cache Hit and Miss, Write Policies, Cache Mapping functions, Cache Page Replacement Algorithms. Virtual Memory, Virtual Memory Page replacement Algorithms, Associative Memory, Memory Interleaving.		**08 Hours**
Module IV	**Input/Output Organization and Communication:** Peripheral Devices, Accessing I/O Devices, I/O Interface, Interrupt. Types of Data Transfer: Parallel and Serial Data Transfer, Synchronous Data Transfer, Asynchronous Data Transfer, Strobe Control,		**04 Hours**

	Handshaking, Asynchronous Serial Transfer. Modes of Transfer: Programmed I/O, Interrupt Initiated I/O, Direct Memory Access (DMA), DMA Controller, I/O Channel & Processor.	
Module V	**Parallel Processing:** Introduction to Pipelining, Instruction Pipeline, Arithmetic Pipeline, Speedup, Efficiency, Throughput, Pipeline Hazards. RISC and CISC Architecture.	**04 Hours**
Essential Readings	1. V. Carl Hamacher, Z. G. Vranesic, and S. G. Zaky, "Computer Organization", TMH. 2. M. Mano, "Computer System Architecture", Prentice Hall of India Pvt. Ltd. / Pearson Education Pvt. Ltd.	
Supplementary Readings	1. William Stallings, "Computer Organization & Architecture", Prentice Hall of India Pvt. Ltd. / Pearson Education Pvt. Ltd. 2. John P. Hayes, "Computer Architecture and Organization", TMH. 3. D. A. Patterson and J. L. Hennessy, "Computer Organization and Design", Morgan Kaufmann Publishers (Elsevier). 4. Kai Hwang and Faye A. Briggs, "Computer Architecture Parallel Processing", TMH.	
Course Outcomes	CO1: Define and memorize different functional units and components of Computer. CO2: Conceptualize and discuss different types of Instruction, Instruction format, Instruction Cycle, Addressing Modes and CPU organization. CO3: Design different types of Memory and CU. CO4: Analyze, compare and differentiate Data transfer techniques. CO5: Solve different Pipeline and Pipeline Hazard problems.	

Design and Analysis of Algorithms

Subject Code:		**Total Contact Hour**	**30**
Semester:	4th Semester	**Total Credit**	**3**
Subject Name:	**DESIGN AND ANALYSIS OF ALGORITHMS**		
Pre-requisites (if any):			
Course Objectives:	1. To understand asymptotic notations to analyze the performance of algorithms 2. To identify the differences in design techniques and apply to solve optimization problems, 3. To apply algorithms for performing operations on graphs and trees, solve novel problems by choosing the appropriate algorithm design technique for their solution. 4. To justify the selection of algorithms 5. To analyze deterministic and nondeterministic algorithms to solve complex problems.		

SYLLABUS		
Module I	Introduction to Design and analysis of algorithms, Asymptotic analysis, Growth of Functions, Asymptotic notations, Recurrences, Solution of Recurrences by substitution, Recursion tree method, Master Method, Brute Force Technique, Divide and Conquer Algorithms, Quicksort, Merge Sort, Binary Search, Strassen's Matrix multiplication, Decrease and Conquer,Heap Sort.	**5**
Module II	Dynamic Programming: Elements of Dynamic Programming, Matrix Chain Multiplication, Longest Common Subsequence, 0/1 Knapsack, Travelling Salesman Problem. Greedy Algorithms: Elements of Greedy Strategy, Activity Selection Problem, Fractional Knapsack Problem, Huffman Codes.	**7**
Module III	Data Structure for Disjoint Sets, Disjoint Set Operations, Minimum Spanning Trees: Kruskal algorithm, Prim's Algorithm, Single Source Shortest paths: Bellmen Ford Algorithm, Dijkstra's Algorithm, All Pair Shortest Path: Floyd-Warsall Algorithm,	**7**
Module IV	String matching: Introduction, Naive string matching algorithm,Rabin-Karp Algorithm, KMP Algorithms, Boyer- Moore Algorithm. Backtracking and Branch and Bound: Introduction, Eight queensproblem , Knapsack problem	**6**
Module V	Introduction to NP completeness: The class P and NP, NP- Complete Problems, NP-Hard Problems, Reduction, Satisfiability and Cook's Theorem, Travelling Salesman problem, Hamiltonian problem, Clique Problem, Approximation algorithms.	**5**
Essential Reading	1.T.H.Coremen, C. E. Leiserson, R.L. Rivest, C. Stein "Introduction to Algorithms"3rd Edition, The MIT Press 2. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, McGraw Hill Education	
Supplementary Reading	1.M.R.Kabat "Design and Analysis of Algorithms", PHI Learning (p) Ltd 2. S. Sridhar "Design and Analysis of Algorithms", Oxford University Press 3. A.V. Aho, J.E. Hopcroft, J.D. Ullman "The Design and Analysis of Algorithms" Pearson Education, NewDelhi 4. K, Louden " Mastering Algorithms", O' Reily Media Inc	
Course Outcomes:	After completion of the course successfully, students will have: **CO1**: Ability of analyzing the performance of algorithms and of finding solution ofdivide and conquer algorithm. **CO2:** Proficiency in finding optimal solutions to various problem **CO3:**Expertise to find MST and shortest path problem. **CO4:**Ability to apply pattern matching algorithms to find particular pattern. **CO5:**Ability to differentiate polynomial and nonpolynomial problems.	

Computer Networks

Subject Code:		**Total Contact Hour**	**30**
Semester:	4th	**Total Credit**	**03**
Subject Name:	**Computer Networks**		
Pre-requisites (if any):	**Basic electronics, Computer Architecture.**		
Course Objectives	1. Introduce the foundational principles and terminology of computer networks, including protocols, architectures, and topologies. 2. Understand how data is transmitted across networks, including concepts like packet-switching, error detection/correction, and analyzes data collision with various protocols 3. Examine multiple accesses, network addressing, and routing in computer network. 4. Apply various routing algorithms over a network to provide optimal path, and examine the addressing entities of a network, study and implementation of transport layer protocols like TCP, UDP. 5. Understand how networks support various applications and services, such as web browsing, email, file sharing, through different protocols		
SYLLABUS			
Module I	**Introduction:** Overview of Data Communications and Networking. Goals of networking, well-known applications such as web, e-mail, need for a layered architecture, OSI and Internet.The physical layer: Basics of communications; Physical media types and their important bandwidth and bit-error-rate characteristics; Wired and Wireless media including copper cables, optical fiber and wireless, Switching Networks.		**06**
Module II	Data Link Layer: Error Detection and correction, Types of Errors, Detection, Error Correction, Data Link Control and Protocols, Flow and Error Control, Stop-and-wait ARQ. Go-Back-N ARQ, Selective Repeat ARQ, HDLC, Point–to-Point Protocol.		**06**
Module III	Multiple Access, Random Access, Controlled Access, Channelization.Local area Network: Ethernet, Traditional Ethernet, Fast Ethernet. Network Layer: Host to Host Delivery, Internetworking, addressing, Routing.		**06**
Module IV	Network Layer Protocols: ARP, RARP, NAT, BOOTP, DHCP, IPV4, ICMP, IPV6, ICMPV6. Transport Layer: Process to Process Delivery: UDP, TCP, congestion control and Quality of service.		**08**
Module V	Application Layer:Client Server Model, Peer to peer network, Domain Name System (DNS): Electronic Mail (SMTP) and file transfer (FTP) HTTP and WWW.		**04**
Essential Readings	1. Data Communications and Networking: Behrouz A. Forouzan, Tata McGraw-Hill, 5th Ed		

	2. Computer Networks: A. S. Tannenbum, D. Wetherall, Prentice Hall, Imprint of Pearson 5th Ed.
Supplementary Readings	1. Computer Networks:A system Approach:Larry L, Peterson and Bruce S. Davie,Elsevier, 4th Ed 2. Computer Networks: Natalia Olifer, Victor Olifer, Willey India 3. Data and Computer Communications: William Stallings, Prentice Hall, Imprint of Pearson, 9th Ed. 4. Data communication & Computer Networks: Gupta, Prentice Hall of India 5. Network for Computer Scientists & Engineers: Zheng, Oxford University Press 6. Data Communications and Networking: White, Cengage Learning
Course Outcomes	CO1: Analyze the concepts of networks, types and architectures CO2: Identify error free transmission of data and analyze data collision with various protocols. CO3: Apply various routing algorithms over a network to provide optimal path. CO4: Examine the addressing entities of a network with implementation of TCP, UDP protocols. CO5: Illustrate the real time applications of networks Protocols

Programming in Python

Subject Code:		**Total Contact Hour**	**30**
Semester:	3rd	**Total Credit**	**3**
Subject Name:	**Programming in Python**		
Pre-requisites (if any):			
Course Objectives:	1: Introduction to Python Language and its features. 2: To understand the concept of Python Program using sequence data and Control statements. 3: To be able to understand and create User Defined Function. 4: To understand the concept of OOPs and its implementation. 5: To understand the concept of strings and file handling		
SYLLABUS			
Module I	**Beginning Python Basics** Introduction to Python Features of Python, Application of Python Data Types, Keywords, Identifiers, Literals, Constants. Python Indentation. Operators and expressions. Naming Conventions with examples, Managing Input and Output, Concept of Indentation. Conditional statements, Looping statements, break and continue, pass & return statements, Nesting of loops.		**6**

Module II	**Modules:** Built-in Modules, Import statement, Packages, Date and Time Modules. Array and its operations, Handling Strings and Characters, List: slicing, bound, cloning, nested list, list and methods, Adding Element: append, extend, count, index and insert). Mutability: Sort, reverse, remove, clear and pop. Map, Filter	**8**
Module III	Tuple and methods, Sets and methods, Dictionary: Basic operation, iterator and methods. **Function:** Introduction to Functions, passing arguments, Anonymous functions (Lambda Function), Recursive Functions.	**6**
Module IV	**Object Oriented Programming:** Classes and Objects, Class methods. Encapsulation, Data Abstraction, Constructor, Destructor and Inheritance. **Exception Handling:** Handling Exceptions: try-except, try-finally	**6**
Module V	**Strings and Regular Expressions :** Methods of String Objects, Escape Sequence, Iterating Strings, String Module, String Formatting, Regular Expressions: Re-Module **File Handling:** Introduction to File Handling, File Operations, Directories.	**4**
Essential Reading	1. Python Programming Python Programming for Beginners By By Adam Stewart 2. Python Cookbook By David Beazley and Brian K. Jones	
Supplementary Reading	1. Introduction to Python Programming By Gowrishankar S. Veena A 2.Python Programming: Using Problem Solving Approach, Oxford University Press by Reema Thareja 3.Python Programming University Press by Ch Satyanarayan, M Radhika, B N Jagadesh	
Course Outcomes:	CO1: Understand the Python Language and its features. CO2: Apply sequence data and control statements to solve problem CO3: Able to create user defined functions to solve problems. CO4: Analyze the concept of OOPs and its implementation. CO5: Create the python program using strings and files.	

Computer Organization Architecture Lab

Subject Code:			**Total Contact Hour**	**20**
Semester:		4th	**Total Credit**	**1.5**
Lab Name:		**Computer Organization and Architecture Lab**		
Course Objectives:		1. Provide students with basic idea of different Combinational and Sequential circuits. 2. Enhance circuit implementation capabilities of students. 3. Familiarize students with the different functional units of the computer. 4. Expertise students with Memory Design and Memory Operations 5. Design of a small project		
Sl. No.	**Expt. No.**	**Experiment Details**		
1	(i)	Design 8-to-1 Multiplexer.		
	(ii)	Design 1-to-8 DeMultiplexer.		
	(iii)	Design 3-to-8 Decoder.		
2		Design BCD to Seven Segment display Decoder.		
3	(i)	Design 8-to-3 Binary Encoder.		
	(ii)	Design 4-to-2 Priority Encoder.		
4		Design 4×3 Array Multiplier.		
5		Design a Universal Shift Register.		
6	(i)	Design Decade Counter.		
	(ii)	Design Up/Down Counter.		
7		Study of PC Trainer and familiarize with different functional units.		
8		Study and test of the ALU Trainer.		
9		Study and test of the Read/Write operation of the Memory unit.		
10		Small Hardware Project design.		
Course Outcomes:		CO1: Define and memorize different functional units and components of Computer. CO2: Design and implement different combinational and sequential circuits. CO3: Test and verify Memory and ALU operations. CO4: Analyze the functionality of different logic circuits. CO5: Solve small problem with circuit design.		

DESIGN AND ANALYSIS OF ALGORITHMS LAB

Subject Code:			**Total Contact Hour**	**20**
Semester:		4th	**Total Credit**	**1.5**
Lab Name:		**Design & Analysis of Algorithms**		
Course Objectives:		• To implement and compare different sorting and searching algorithms. • To demonstrate the implementation of various divide and conquer algorithms. • To find optimal solution using dynamic and greedy algorithm. • To implement spanning tree and shortest path algorithm • To Solve string matching problem		
Sl.	**Expt.**	**Experiment Details**		

No.	No.	
1	1	**Illustration of Analysis of Algorithms:** Comparison of sorting algorithms like Bubble, Insertion and Selection. Heap sort using a max heap.
2	2	**Divide and Conquer Algorithm:** i. Quick Sort, ii. Merge Sort iii. Binary Search.
3	3	**Application:** i. Quick sort ii. Merge Sort iii. Binary Search.
4	4	**Dynamic Programming:** i. Longest Common Subsequence Problem ii. Matrix Chain Multiplication Problem iii.0/1 knapsack Problem iv. Travelling Salesman Problem
5	5	**Greedy Algorithm:** i. Fractional knapsack problem ii. Huffman Coding
6	6	**Minimum Spanning Tree:** i. Kruskal's algorithm ii. Prim's algorithm
7	7	**Shortest Path Problem:** i. Dijkstra algorithm ii. Bellman ford algorithm
8	8	**Backtracking and Branch & Bound:** Queen Problem
9	9	**String Marching Algorithm** i. Naive string matching algorithm ii. Rabin karp algorithm
10	10	**Approximation Algorithm:** Travelling Salesman Problem
Course Outcomes:		CO1: Ability to compare and implement various divide and conquer algorithms. CO2: Improved skill in choosing and developing algorithms for optimization problems. CO3: Potential to demonstrate and implement graph algorithms like spanning tree and shortest path. CO4: Ability to implement different string matching algorithms. CO5: Using approximation algorithm for NP complete problems.

Computer Networks Lab

Subject Code:		**Total Contact Hour**	**20**
Semester:	4th	**Total Credit**	**1.5**
Lab Name:	**Computer Networks Lab**		
Course Objectives:	The objective of this lab course is to: 1. To understand the working principle of various communication protocols.		

		2. To know the concept of data transfer between nodes. 3. To analyze the various routing algorithms. **4.** Analyze structure and formats of TCP/IP layer protocols using network tools. **5.** Implementing various network algorithms such as error control, error detection, routing, and security related algorithms.
Sl. No.	**Expt. No.**	**Experiment Details**
1		Introduction to Packet Tracer and Implementation of different Network Topology using Packet Tracer.
2	i.	Limited broadcast and directed broadcast.
	ii	IP addressing with class full and class less addressing scheme.
	iii	Sub netting and super netting.
	iv	Concept of CIDR.
3		Assigning static IP address to PC and implement basic command of Computer network like PING, traceroute etc.
4		Implementing VLSM network using Packet Tracer.
5		Understanding Router concept, types of router, different type of ports on router and how to configure a Router.
6		Configure network topology and implement static routing using Packet Tracer.
7		Configure network topology and implement dynamic routing protocol such as RIP, EIGRP etc. using Packet Tracer.
8	i	Configure DHCP Server in the Network using packet tracer.
	ii	Configure HTTP Server in the Network using packet tracer.
	iii	Configure DNS Server in the Network using packet tracer.
9		Implementation of VLANS using packet tracer.
10		Troubleshooting existing network.
Course Outcomes:		1. Identify and use various networking components, transmission media for establishing a network 2. Implement n/w topology using network devices through packet tracer. 3. Analyze performance of various communication protocols. 4. Understand and configure routing algorithms through packet tracer. 5. Implement device sharing and troubleshooting in the network.

Programming in Python Lab

Subject Code:		**Total Contact Hour**	**20**
Semester:	3rd	**Total Credit**	**1.5**
Lab Name:	**Programming in Python Lab**		
Course Objectives:	1: Introduction to Python Language and its features. 2: To understand the concept of Python Program using sequence data and Control statements. 3: To be able to understand and create User Defined Function. 4: To understand the concept of OOPs and its implementation. 5: To understand the concept of strings and file handling		

Sl. No.	Expt. No.	Experiment Details
1		Program on basics of python Programming Language.
2		Program on basic Data Structures in Python.
3		Program on Conversion from on data type to another.
4		Program on Functions in Python.
5		Program using Object Oriented Programming in Python.
6		Program using Inheritance in Python.
7		Program using String in Python.
8		Program using Regular expression in Python.
9		Program using File Handling in Python.
10		Program using basics of Pandas and Matplotlib module in Python.
Course Outcomes:		CO1: Understand the Python Language and its features. CO2: Apply sequence data and control statements to solve problem CO3: Able to create user defined functions to solve problems. CO4: Analyze the concept of OOPs and its implementation. CO5: Create the python program using strings and files.