Gurugram University Gurugram

Curriculum for UG Degree Course in

Computer Science Engineering - (Internet of Things and Cyber security Including Blockchain Technology)

B. Tech. (Computer Science Engineering - (Internet of Things and Cyber security Including Blockchain Technology))
Scheme of Studies/Examination w.e.f. 2023-24

Semester - III

S.No.	Category	Course Code	Course Title	Hours Per week			Total Contact Hrs. per week	Credits	Examination Schedule (Marks)			
				L	T	P			Marks of classwork	Theory	Practical	Total
1.	**ESC**		Digital Electronics	3	0	0	3	3	30	70		100
2.	**PCC**		Advanced-Data Structure	3		0	3	3	30	70		100
3.	**PCC**		Database Management Systems with SQL	3	0	0	3	3	30	70		100
4.	**PCC**		Programming with C++	3	0	0	3	3	30	70		100
5.	**PCC**		Foundations of Blockchain Technology	3	0	0	3	3	30	70		100
6.	**BSC**		Mathematics - III	3	0	0	3	3	30	70		100
7.	**LC**		Digital Electronics Lab	0	0	2	2	1	50		50	100
8.	**LC**		Advanced-Data Structure Lab	0	0	2	2	1	50		50	100
9.	**LC**		Database Management Systems Lab	0	0	2	2	1	50		50	100
10.	**LC**		Programming with C++ Lab	0	0	2	2	1	50		50	100
11.	**MC**		Constitution of India*	2	0	0	2	-	30	70		100*
			Total	**20**	**0**	**8**	**28**	**22**	**380**	**420**	**200**	**1000**

***NOTE:** The examination of the regular students will be conducted by the concerned college/Institute internally. Each student will be required to score a minimum of 40% marks to qualify in the paper. The marks will not be included in determining the percentage of marks obtained for the award of a degree. However, these marks will be shown in the detailed marks certificate of the students.

DIGITAL ELECTRONICS

Semester	III				
Course code					
Category	Engineering Science courses				
Course title	Digital Electronics				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To understand the basic theoretical concepts of digital systems like the binary system and Boolean algebra.
2. To use Boolean algebraic formulations to design digital systems. To design using combinational/sequential circuits**.**
3. To express real-life problems in logic design terminology.
4. To understand the logic of adders, subtractors and converters.

UNIT - I

FUNDAMENTALS OF DIGITAL SYSTEMS AND LOGIC FAMILIES

Digital signals, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive-OR operations, Boolean algebra, examples of IC gates, number systems - binary, signed binary, octal hexadecimal number, binary arithmetic, one's and two's complements arithmetic, codes, error detecting and correcting codes.

UNIT - II

COMBINATIONAL DIGITAL CIRCUITS

Standard representation for logic functions, K-map representation, and simplification of logic functions using K-map, minimization of logical functions. Don't care conditions, Multiplexer, De-Multiplexer, Decoders, Adders, Subtractors, BCD arithmetic, carry look-ahead adder, serial adder, ALU, elementary ALU design, popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, Q-M method of function realization.

UNIT - III

SEQUENTIAL CIRCUITS AND SYSTEMS

A 1-bit memory, the circuit properties of the Bistable latch, the clocked SR flip-flop, J-K flip-flop, T flip-flop and D flip-flop, applications of flip-flops, shift registers, applications of shift registers, serial-to-parallel converter, parallel-to-serial converter, ring counter, sequence generator, ripple

(Asynchronous) counters, synchronous counters, counters design using flip-flops, special counter IC's, asynchronous sequential counters, applications of counters**.**

UNIT - IV

A/D AND D/A CONVERTERS

Digital to analog converters: weighted resistor/converter, R-2-R Ladder D/A converter, specifications for D/A converters, examples of D/A converter ICs, sample and hold circuit, Analog to digital converters: quantization and encoding, parallel comparator A/D converter, successive approximation A/D converter, counting A/D converter, dual slope A/D converter.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Outline the general concepts and terminology related to logic gates, logic families, combinational and sequential circuits.
CO2: Discuss the basic analog/digital components and their interconnections in logic families and circuits.
CO3: Apply different methods/techniques to design various digital circuits.
CO4: Analyse day to day problems and industrial problems for their solutions using digital circuits.
CO5: Contrast different types of digital circuits and their designing methods.
CO6: Design digital circuit for various practical problems.

TEXT AND REFERENCE BOOKS:

1. R. P. Jain, "Modern Digital Electronics", McGraw Hill Education, 2009.
2. M. M. Mano, "Digital logic and Computer Design", Pearson Education India,2016.
3. A. Kumar, "Fundamentals of Digital Circuits", Prentice Hall India, 2016.
4. Nasib Singh Gill and J B Dixit, "Digital Design and Computer Organization", University Science Press, New Delhi

ADVANCE DATA STRUCTURE

Semester	III				
Course code					
Category	Professional Core Courses				
Course title	Advance Data Structure				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To understand and apply linear data structures-List, Stack and Queue.
2. To understand the tree algorithms and their applications.
3. To learn different algorithms and analysis techniques.
4. To apply sorting algorithms in real-time applications

UNIT - I

Review of Linear Data Structures

Linked List: Traverse, Insertion, Deletion; Circular List: Traverse, Insertion, Deletion, Doubly List and Circular List: Insertion and deletions; Stacks and Queue implementation using linked list.

Introduction to Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries.

Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, and Extendible Hashing.

UNIT - II

Advanced Trees: Trees: Review of binary trees and binary search trees: traversing, insertion, and deletion; **AVL Trees:** Introduction to AVL trees, Rotations in AVL trees: LL, RR, LR & RL,

Introduction, Search, Insert & delete operations: Red-Black Trees, 2-3 Trees, B-Trees, B+ Trees, Splay Trees.

UNIT - III

Sets: Representation on Sets, Operations on Sets, Application on Sets,

Files: File Concepts, File organization, Files and Streams, Working with Files Using I/O Stream, Sequential File Organization, Direct File Organization, Indexed Sequential Organization

UNIT - IV

Graphs: Representation, Basic terminology, traversal, connected components, shortest path, topological sort, Dijkstra's Algorithm, Floyd Warshall's Algorithm, network flow problems.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Design and Analyze programming problem statements.
CO2: Understand the ADT/libraries, and use it to design algorithms for a specific problem.
CO3: Select algorithm design approaches in a problem-specific manner.
CO4: Compare & contrast the complexity analysis of various sorting & searching algorithms.
CO5: To be able to analyse the efficiency of algorithms.
CO6: Implement various data structure concepts on real-world industrial problems.

TEXT AND REFERENCE BOOKS:

1. Seymour Lipschutz: Data Structures with C, Schaum's outline by TMH
2. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.
3. R.B. Patel: Expert Data Structures in C, Khanna Publishers,2001.
4. R.L. Kruse: Data Structures & Program Design in C, PHI.
5. D.F. Knuth: The art of Computer Programming Vol 1, Narosa Publications,1985.
6. Byron S. Gottfried & J K Chhabra: Theory and Problems of Programming with C Language, Schaum's Series, TMH,2005.

DATABASE MANAGEMENT SYSTEMS WITH SQL

Semester	III				
Course code					
Category	Professional Core Courses				
Course title	Database Management Systems With SQL				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To understand the different issues involved in the design and implementation of a database system.
2. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
3. To understand and use data manipulation language to query, update, and manage a database
4. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

UNIT - I

Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object-oriented data models, integrity constraints, data manipulation operations.

UNIT - II

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

UNIT - III

Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp-based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

Storage strategies: Indices, B-trees, hashing,

UNIT - IV

Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

Advanced topics: Object-oriented and object-relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: For a given query, write relational algebra expressions for that query and optimize the developed expressions

CO2: For a given requirement specification, design the databases using E R method and normalization.

CO3: For a given specification, construct the SQL queries for Open source and Commercial DBMS - MYSQL, ORACLE, and DB2.

CO4: For a given query, optimize its execution using Query optimization algorithms

CO5: For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.

CO6: Implement the isolation property, including locking, and time stamping based on concurrency control and Serializability of scheduling.

TEXT AND REFERENCE BOOKS:

1. Database System Concepts, 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill
2. Principles of Database and Knowledge–Base Systems, Vol 1 by J. D. Ullman, Computer Science Press.
3. Fundamentals of Database Systems, 5th Edition by R. Elmasri and S. Navathe, Pearson Education
4. Foundations of Databases, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

PROGRAMMING WITH C++

Semester	III				
Course code					
Category	Professional Core Courses				
Course title	Programming with C++				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To understand how the C++ is superset of C by incorporating the Object-oriented features in C language.
2. To learn how to efficiently use the memory using Pointers and Dynamic Memory Management
3. To learn how to implement different types of constructors and the use of destructor.
4. To learn how to implement the concept of data abstraction, encapsulation and how to perform different types of overloading i.e. operators and functions.
5. To learn how inheritance helps to reuse the code.
6. To learn how we can implement dynamic binding with polymorphism.
7. To learn the use of exception handling in C++ programs.

UNIT - I

Object-Oriented Programming Concepts: Introduction, comparison between procedural programming paradigm and object-oriented programming paradigm, basic concepts of object-oriented programming — concepts of an object and a class, interface and implementation of a class, operations on objects, relationship among objects, abstraction, encapsulation, data hiding, inheritance, overloading, polymorphism, messaging.

Classes and Objects: Specifying a class, creating class objects, accessing class members, access specifiers, static members, use of const keyword, friends of a class, empty classes, nested classes, local classes, abstract classes, container classes, bit fields and classes.

UNIT - II

Inheritance: Introduction, defining derived classes, forms of inheritance, ambiguity in multiple and multipath inheritances, virtual base class, object slicing, overriding member functions, object composition and delegation, order of execution of constructors and destructors.

Pointers and Dynamic Memory Management: Declaring and initializing pointers, accessing data through pointers, pointer arithmetic, memory allocation (static and dynamic), dynamic memory management using new and delete operators, the pointer to an object, this pointer, pointer related problems - dangling/wild pointers, null pointer assignment, memory leak and allocation failures.

UNIT - III

Constructors and Destructors: Need for constructors and destructors, copy constructor, dynamic constructors, explicit constructors, destructors, constructors and destructors with static members, initializer lists.

Operator Overloading and Type Conversion: Overloading operators, rules for overloading operators, overloading of various operators, type conversion - basic type to class type, class type to basic type, class type to another class type.

Virtual functions & Polymorphism: Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract clasess, virtual destructors.

UNIT - IV

Exception Handling: Review of traditional error handling, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions.

Templates and Generic Programming: Template concepts, Function templates, class templates, illustrative examples.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Understand the concept of Object-Oriented Programming through C++.

CO2: Identify importance of object-oriented programming and difference between Procedural programming and object oriented programming features.

CO3: be able to make use of objects and classes for developing programs.

CO4: be able to use various object-oriented concepts to solve different problems.

CO5: be able to develop the programs /Projects using some advanced features of C++ Programming.

TEXT AND REFERENCE BOOKS:

1. BjraneStroustrup, “C++ Programming language”,3rd edition, Pearson education Asia(1997)
2. LaforeR.”Object oriented Programming in C++”,4th Ed. Techmedia,New Delhi(2002).
3. Yashwant Kenetkar,”Let us C++”,1stEd.,Oxford University Press(2006)
4. B.A. Forouzan and R.F. Gilberg,CompilerScience,”A structured approach using C++” Cengage Learning, New Delhi.

MATHEMATICS - III

Semester	III				
Course code					
Category	Basic Science Courses				
Course title	Mathematics - III				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. Understand the basic concepts of Limits & Continuity.
2. Understand partial and total derivative and apply it to real time applications.
3. Understand the ordinary Differential Equations of First, Second and Higher Order.

UNIT - I

Multivariable Differential Calculus: Limit, Continuity and Partial derivatives, Homogeneous functions, Euler's Theorem, Total derivative, Maxima, Minima and Saddle points, Lagrange's method of undetermined multipliers.

UNIT - II

Multivariable Integral Calculus: Double integral, change of order of integration, Change of variables, Applications of double integral to find area enclosed by plane curves, Triple integral.

UNIT - III

Ordinary Differential Equations of first order: Linear and Bernoulli's equations, Exact differential equations, Equations reducible to exact differential equations, Applications of differential equations of first order and first degree to simple electric circuits, Newton's law of cooling, Heat flow and Orthogonal trajectories.

UNIT - IV

Ordinary Differential equations of second and higher order: Linear differential equations of second and higher order, Complete solution, Complementary function and Particular integral, Method of variation of parameters to find particular integral, Cauchy's and Legendre's linear equations, Simultaneous linear differential equations with constant coefficients, Applications of linear differential equations to oscillatory electric circuits.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Deal with functions of several variables and evaluate partial derivative.

CO2: Evaluate multiple integrals and their usage.

CO3: Solve ordinary differential equations that model physical processes.

CO4: Formulate and solve problems involving moment of inertia, volume and centre of gravity.

CO5: Solve engineering problems related to oscillatory electric circuits.

CO6: Solve field problems in engineering involving Ordinary Differential Equations like R-L-C circuits and to find heat loss

TEXT AND REFERENCE BOOKS:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, Pearson Education.
2. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons.
3. B. V. Ramana, Higher Engineering Mathematics, Tata McGraw-Hill Publishing Company Limited.
4. N. P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications.
5. B. S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
6. W. E. Boyce and R. C. DiPrima, Elementary Differential Equations and Boundary Value Problems,Wiley India.
7. S. L. Ross, Differential Equations, Wiley India.
8. E. A. Coddington, An Introduction to Ordinary Differential Equations, Prentice Hall India.
9. E. L. Ince, Ordinary Differential Equations, Dover Publications

FOUNDATIONS OF BLOCKCHAIN TECHNOLOGY

Semester	III				
Course code					
Category	Professional Core Courses				
Course title	Foundations of Blockchain Technology				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To introduce basic concepts of Blockchain.
2. To understand abstract models for Blockchain technology
3. To learn about usage of Blockchain technology in financial services.
4. To visualize the scope of blockchain & its role in futuristic development.

UNIT - II

Introduction to Blockchain: Overview of blockchain, need for blockchain, history of centralized services, trusted third party, Distributed consensus in open environments, Distributed Vs Decentralized Network, 51 % attack theory, Public blockchains, Private blockchains, Blockchain Architecture and working, Mining, Limitations of blockchain, Applications of blockchain

UNIT - II

Models for blockchain: GARAY model, RLA Model, Proof of Work (PoW), HashcashPoW, PoW Attacks and the monopoly problem, Proof of Stake(PoS), hybrid models(PoW+PoS), Proof of Burn and Proof of Elapsed Time.

UNIT - III

Permissioned Blockchain: Permissioned model and use cases, Design issues for Permissioned blockchains, State machine replication, Consensus models for permissioned blockchain, Distributed consensus in closed environment, Paxos, RAFT Consensus, Byzantine general problem, Byzantine fault tolerant system, Lamport-Shostak-Pease BFT Algorithm, BFT over Asynchronous systems.

UNIT - IV

Blockchain in Financial Service:- Digital Currency, Cross border payments, Steller and Ripple protocols, Project Ubin, Know Your Customer (KYC), Privacy Consents, Mortgage over Blockchain, Blockchain enabled Trade, We Trade – Trade Finance Network, Supply Chain Financing, Insurance.

Blockchain Security: Security properties, Security considerations for Blockchain, Intel SGX, Identities and Policies, Membership and Access Control, Blockchain Crypto Service Providers, Privacy in a Blockchain System, Privacy through Fabric Channels, Smart Contract Confidentiality.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Recognizing goals of Blockchain.

CO2: Understanding the basics of Blockchain, notation of Distributed Systems in Blockchain, and analyzing various problems.

CO3: Smart Contracts, transactions in Blockchain and Permissioned Blockchain.

CO4: Analyzing usage of Blockchain in finance.

CO5: Security issues in Blockchain.

TEXT AND REFERENCE BOOKS:

1. Blockchain: Blueprint for a New Economy, by Melanie Swan.
2. Blockchain: The blockchain for beginner's guide to blockchain technology and leveraging blockchain programming, by Josh Thompsons
3. Blockchain Basics by Daniel Drescher, Apress

CONSTITUTION OF INDIA

Semester	III				
Course code					
Category	Mandatory courses				
Course title	Constitution of India				
Scheme and Credits	L	T	P	Credits	
	2	0	0	-	
Classwork	30				
Exam	70				
Total	100				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
2. To address the growth of Indian opinion regarding modern Indian intellectuals' constitutional role and entitlement to civil and economic rights as well as the emergence of nationhood in the early years of Indian nationalism.
3. To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.
4. To learn procedure and effects of emergency, composition and activities of election commission and amendment procedure.

UNIT - I

Philosophy of Indian Constitution: Salient features of Indian Constitution, Preamble, and Nature of Indian Constitution, Procedure for amendment of the Constitution.

UNIT - II

Federal structure and distribution of legislative and financial powers between the Union and the States

UNIT - III

Organs of Governance: President – Qualification and Powers of the President, Governor- Qualification and Powers of Governor,

Parliament: Composition, Qualifications and Disqualifications, Judiciary: Appointment, Tenure and Removal of Judges.

UNIT - IV

Fundamental Rights: Origin and development of Fundamental rights, Need for fundamental rights. Introduction to Rights to equality, right to freedom, right against exploitation, Right to freedom of religion, Cultural and Education rights and Fundamental duties.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.

CO2: Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to a revolution in India.

CO3: Exercise his fundamental rights in proper sense at the same time identifies his responsibilities in national building.

CO4: Discuss the circumstances surrounding the foundation of the Congress Socialist Party [CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution.

CO5: Discuss the passage of the Hindu Code Bill of 1956.

CO6: Analyse the Indian political system, the powers and functions of the Union, State and Local Governments in detail.

TEXT AND REFERENCE BOOKS:

1. The Constitution of India, 1950 (Bare Act), Government Publication.
2. Dr. S.N. Busi, Dr. B.R. Ambedkar framing of Indian Constitution, latest Edition
3. M.P. Jain, Indian Constitution Law, Lexis Nexis, latest edition
4. D.D. Basu, Introduction to Constitution of India, Lexis Nexis, latest edition.

NOTE: The examination of the regular students will be conducted by the concerned college/Institute internally. Each student will be required to score a minimum of 40% marks to qualify in the paper. The marks will not be included in determining the percentage of marks obtained for the award of a degree. However, these marks will be shown in the detailed marks certificate of the students.

DIGITAL ELECTRONICS LAB

Semester	III				
Course code					
Category	Laboratory course				
Course title	Digital Electronics Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

Implementation of all experiments with the help of Bread-Board.

1. Study of Logic Gates: Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates; Realization of OR, AND, NOT and XOR functions using universal gates.
2. Half Adder / Full Adder: Realization using basic and XOR gates.
3. Half Subtractor / Full Subtractor: Realization using NAND gates.
4. 4-Bit Binary-to-Gray & Gray-to-Binary Code Converter: Realization using XOR gates.
5. 4-Bit and 8-Bit Comparator: Implementation using IC7485 magnitude comparator chips.
6. Multiplexer: Truth-table verification and realization of Half adder and Full adder.
7. Demultiplexer: Truth-table verification and realization of Half subtractor and Full subtractor.
8. Flip Flops: Truth-table verification of JK Master Slave FF, T-type and D-type FF.
9. Asynchronous Counter: Realization of 4-bit up counter and Mod-N counter.
10. Synchronous Counter: Realization of 4-bit up/down counter and Mod-N counter.
11. Shift Register: Study of shift right, SIPO, SISO, PIPO, PISO & Shift left operations.
12. DAC Operation: Study of 8-bit DAC, obtain staircase waveform.
13. ADC Operations: Study of 8-bit ADC

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Define different types of logic gates, identify their ICs and also verify their truth table.

CO2: Derive basic logic gates, adder, and subtractor using universal gates.

CO3: Illustrate realization of Boolean expression in SOP and POS form and design it using logic gates.

CO4: Design and test combinational circuits.

CO5: Design and develop sequential circuits.

CO6: Demonstrate team-based laboratory activities with fellow students to interact effectively on a social and interpersonal level.

DATABASE MANAGEMENT SYSTEM LAB

Semester	III				
Course code					
Category	Laboratory course				
Course title	Database Management System Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

1. Design a Database and create required tables. For e.g. Bank, College Database
2. Apply the constraints like Primary Key, Foreign key, NOT NULL to the tables.
3. Write a SQL statement for implementing ALTER, UPDATE and DELETE.
4. Write the queries to implement the joins.
5. Write the query for implementing the following functions: MAX (), MIN (), AVG () and COUNT ().
6. Write the query to implement the concept of Integrity constrains.
7. Write the query to create the views.
8. Perform the queries for triggers.
9. Perform the following operation for demonstrating the insertion, updation and deletion.
10. Using the referential integrity constraints.
11. Write the query for creating the users and their role. Mini project (Application Development using Oracle/ MySQL)
    a. Inventory Control System
    b. Material Requirement Processing.
    c. Hospital Management System.
    d. Railway Reservation System.
    e. Personal Information System.
    f. Web-Based User Identification System.
    g. Time Table Management System.
    h. Hotel Management

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Identify the fundamental elements of relational database management systems.

CO2: Design and explain the basic concepts of relational data model, entity-relationship model, and relational database design.

CO3: Apply the relational database theory to formulate basic and advanced SQL queries and relational algebra expressions for the queries.

CO4: Identify the use of normalization and functional dependency in database design.

CO5: Understand the concept of transactions and serializability in database management system.

CO6: Classify the implementation details of Concurrency control protocols and discuss various database recovery methods.

ADVANCED DATA STRUCTURE LAB

Semester	III				
Course code					
Category	Laboratory course				
Course title	Advance Data Structure Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

1. Write a program to implement all operations on 1-D array.
2. Write a program to implement all operations on Simple Linked List.
3. Write a program to implement all operations on a circular Linked List.
4. Write a program to implement all operations on a doubly Linked List.
5. Write a program to implement all operations on a doubly circular Linked List.
6. Write a program to implement all operations on Stack using Array.
7. Write a program to implement all operations on Stack using Linked List.
8. Write a program to implement all operations on Queue using Array.
9. Write a program to implement all operations on Queue using Linked List.
10. Write a Program to implement dictionary techniques.
11. Write a program to implement hashing techniques.
12. Write a Program to implement Red-Black Trees.
13. Write a Program to implement Binary Search Trees.
14. Write a Program to design a menu to implement: Quick, Merge, and Bubble sorting.
15. Write a Program to develop a recursive Program to implement Breadth First Search and Depth First Search.
16. Write a Program to develop a non-recursive Program to implement Breadth First Search and Depth First Search

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Identify the appropriate data structure for a given problem.
CO2: Implement Dictionary by using hashing techniques.
CO3: Analyse various basic operations of trees to improve the efficiency.
CO4: Build a Binary Heap using Priority queues.
CO5: Apply the concepts of data structures in various real-world applications.
CO6: Identify, model, solve and develop algorithms for real-life problems like shortest path and MST using graph theory.

PROGRAMMING WITH C++ LAB

Semester	III				
Course code					
Category	Laboratory course				
Course title	Programming with C++ Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

1. Write a program that uses a class where the member functions are defined inside a class.
2. Write a program that uses a class where the member functions are defined outside a class.
3. Write a program to demonstrate the use of static data members.
4. Write a program to demonstrate the use of const data members.
5. Write a program to demonstrate the use of zero argument and parameterized constructors.
6. Write a program to demonstrate the use of dynamic constructor.
7. Write a program to demonstrate the use of explicit constructor.
8. Write a program to demonstrate the use of initializer list.
9. Write a program to demonstrate the overloading of increment and decrement operators.
10. Write a program to demonstrate the overloading of binary arithmetic operators.
11. Write a program to demonstrate the overloading of memory management operators.
12. Write a program to demonstrate the multilevel inheritance.
13. Write a program to demonstrate the multiple inheritance.
14. Write a program to demonstrate the virtual derivation of a class.
15. Write a program to demonstrate the runtime polymorphism.
16. Write a program to demonstrate the exception handling.
17. Write a program to demonstrate the use of function template.
18. Write a program to demonstrate the use of class template.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Understand dynamic memory management techniques using pointers, constructors, destructors, etc.

CO2: Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.

CO3: Classify inheritance with the understanding of early and late binding

CO4: Usage of exception handling and generic programming.

CO5: Develop the programs /Projects using some advanced features of C++ Programming.

CO6: Percept the utility and applicability of OOP.

4TH SEMESTER

B.Tech. (Computer Science Engineering - (Internet of Things and Cyber security Including Blockchain Technology))
Scheme of Studies/Examination w.e.f. 2023-24

Semester - IV

S.N.	Category	Course Code	Course Title	Hours Per week			Total Contact Hrs. per week	Credits	Examination Schedule (Marks)			
				L	T	P			Marks of classwork	Theory	Practical	Total
1.	**PCC**		Operating System	3	0	0	3	3	30	70		100
2.	**PCC**		Introduction to Internet of Things	3	0	0	3	3	30	70		100
3.	**PCC**		Programming in Java	3	0	0	3	3	30	70		100
4.	**PCC**		Microprocessor and Micro-controller	3	0	0	3	3	30	70		100
5.	**BSC**		Discrete Mathematics	3	0	0	3	3	30	70		100
6.	**PCC**		Computer Organization & Architecture	3	0	0	3	3	30	70		100
7.	**LC**		Operating System Lab	0	0	2	2	1	50		50	100
8.	**LC**		Programming in Java Lab	0	0	2	2	1	50		50	100
9.	**LC**		Microprocessor and Micro-controller Lab	0	0	2	2	1	50		50	100
10.	**LC**		Internet of Things Lab Lab	0	0	2	2	1	50		50	100
			Total	**18**	**0**	**8**	**26**	**22**	**380**	**420**	**200**	**1000**

NOTE: At the end of 4th semester each student has to undergo Practical Training of 4/6 weeks in an Industry /Institute/ Professional Organization/Research Laboratory/ training centre etc. and submit the typed report along with a certificate from the organization & its evaluation shall be carried out in the 5th Semester.

OPERATING SYSTEM

Semester	IV				
Course code					
Category	Professional Core Courses				
Course title	Operating System				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To understand the mechanisms of OS to handle processes and threads and their communication.
2. To understand the process management mechanisms and scheduling algorithms.
3. To understand the mechanisms involved in memory management in OS and virtual memory concepts.
4. To understand the file management and deadlocks handling techniques in OS.

UNIT - I

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services.

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Multithreading.

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SJF, SRTF, RR Scheduling.

UNIT - II

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, and Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

UNIT - III

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation

and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Optimal Page Replacement and Least Recently used (LRU).

UNIT - IV

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), efficiency and performance.

Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks. Case study on UNIX and WINDOWS Operating System.

Case Studies: Comparative study of WINDOW, UNIX & LINUX system.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Explain the basic concepts of operating system.
CO2: Describe mechanisms of OS to handle processes, threads, and their communication.
CO3: Analyze the memory management and its allocation policies.
CO4: Illustrate different conditions for deadlock and their possible solutions.
CO5: Discuss the storage management policies with respect to different storage management technologies.
CO6: Evaluate the concept of the operating system with respect to UNIX, Linux, Time, and mobile OS.

TEXT AND REFERENCE BOOKS:

1. Operating System Concepts Essentials, 9th Edition by AviSilberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.
3. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
4. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
5. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
6. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Introduction to Internet of Thing

Semester	IV				
Course code					
Category	Professional Core Courses				
Course title	Introduction to Internet of Things				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. Understand internet of Things and its hardware and software components.
2. Interface I/O devices, sensors & communication modules.
3. Remotely monitor data and control devices.
4. Develop real life IoT based projects.

UNIT - I

Introduction to IOT: Vision, Definition, Conceptual Framework, Architectural view, technology behind IoT, Sources of the IoT, M2M Communication, IoT Examples. Design Principles for Connected Devices: IoT/M2M systems layers and design standardization, communication technologies, Business processes in IoT, Everything as a Service (XaaS), Role of Cloud in IoT, Security aspects in IoT.

UNIT - II

Elements of IOT

Hardware Components- Sensors, Digital sensors, actuators, radio frequency identification (RFID) technology, wireless sensor networks, participatory sensing technology. **Embedded Platforms for IOT:** Embedded computing basics, Overview of IOT supported Hardware platforms such as Arduino, NetArduino, Raspberry pi, Beagle Bone, Intel Galileo boards and ARM cortex

Software Components- Programming API's (using Python/Node.js/Arduino) for Communication Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP.

UNIT - III

Network & Communication aspects in IoT: Wireless Medium access issues, MAC protocol survey, Survey routing protocols, Sensor deployment & Node discovery, Data aggregation & dissemination.

Programming the Ardunio: Ardunio Platform Boards Anatomy, Ardunio IDE, coding, using emulator, using libraries, additions in ardunio, programming the ardunio for IoT.

UNIT - IV

Challenges in IoT Design challenges: Development Challenges, Security Challenges, Other challenges IoT Applications: Smart Metering, E-health, City Automation, Automotive Applications, home automation, smart cards, communicating data with H/W units, mobiles, tablets, Designing of smart street lights in smart city.

IoT Case Studies: IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Demonstrate basic concepts, principles and challenges in IoT.
CO2: Illustrate functioning of hardware devices and sensors used for IoT.
CO3: Analyze network communication aspects and protocols used in IoT.
CO4: Apply IoT for developing real life applications using Ardunio programming.
CO5: To develop IoT infrastructure for popular applications.

TEXT AND REFERENCE BOOKS:

1. Olivier Hersent, David Boswarthick, Omar Elloumi “The Internet of Things key applications and protocols”, willey
2. Jeeva Jose, Internet of Things, Khanna Publishing House
3. Michael Miller “The Internet of Things” by Pearson
4. Raj Kamal “INTERNET OF THINGS”, McGraw-Hill, 1ST Edition, 2016
5. Arshdeep Bahga, Vijay Madisetti “Internet of Things (A hands on approach)” 1ST edition, VPI publications,2014
6. Adrian McEwen,Hakin Cassimally “Designing the Internet of Things” Wiley India

PROGRAMMING IN JAVA

Semester	IV				
Course code					
Category	Professional Core Courses				
Course title	Programming in Java				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methos etc.
2. Be able to use the Java SDK environment to create, debug and run simple Java programs.
3. To analyze the object-oriented paradigm using java programming language.
4. To implement small/medium scale java programs to resolve small business problems.

UNIT - I

Introduction to Java: Evolution of Java, Object Oriented Programming Structure, Overview and characteristics of Java, Java program Compilation and Execution Process, Organization of the Java Virtual Machine, Client side Programming, Platform Independency & Portability, Security, Relation b/w JVM, JRE and JDK, Introduction to JAR format, Naming Conventions, Data types & Type casting, operators, Security Promises of the JVM, Security Architecture and Security Policy, security aspects, sandbox model.

UNIT - II

OOPS Implementation: Classes, Objects, attributes, methods, data encapsulation, reference variables, Constructors, Anonymous block, Method Overloading, Static Data members, Block & methods; Memory Structure: Stack, Heap, Class & Method area

Class loading & Execution flow: Static vs Dynamic Class loading, implicit vs explicit class loading, class loading operations;

Argument Passing Mechanism: Passing primitive arguments, passing objects, Wrapper Classes;

This keyword: Referencing instance members, Intra class constructor chaining, Method chaining;

Inheritance & code reusability: Extending classes for code reusability, Usage of super keyword, Method Overriding, Object class

Inheritance & Runtime Polymorphism: Static & Dynamic binding, Inheritance and Is-A relation, Runtime Polymorphism and Generalization, Abstract classes & methods, Final Keyword;

Interfaces and Role based Inheritance: Feature & Role based Inheritance, Static & Dynamic classing Environment, classes & interfaces, interface applications in real scenarios; Has-A relation:

Aggregation & Composition, Nested classes, Inner classes, Anonymous Inner classes, String Buffer Class, tokenizer, applets, Life cycle of applet and Security concern

UNIT - III

Threads: Creating Threads, Thread Priority, Blocked States, Extending Thread Class, Runnable Interface, Starting Threads, Thread Synchronization, Synchronize Threads, Sync Code Block, Overriding Synced Methods, Thread Communication, wait, notify and notify all.

Swing & AWT:

Swing class hierarchy, containers, user interface components, graphics context, AWT Components, Component Class, Container Class, Layout Manager Interface Default Layouts, Insets and Dimensions, Border Layout, Flow Layout, Grid Layout, Card Layout Grid Bag Layout AWT Events, Event Models, Listeners, Class Listener, Adapters, Action Event Methods Focus Event Key Event, Mouse Events, Window Event

Package & Scopes: Need of Packages, associating classes to Packages, Class path environment variable, Import Keyword and Feature of static import, Public, protected, private & default scope, Private Inheritance;

Exception Handling: exception and error, Exception Handling & Robustness, Common Exceptions and Errors, Try and catch block, Exception handlers, throw keyword, Checked and Unchecked Exceptions, Role of finally, User defined Exceptions.

UNIT - IV

Collection Framework: Role and Importance of Collection Framework, List & Set based collection, Iterator & List Iterator, Maps, Searching elements in List, Hash and Tree based collections, Role of equals and hashCode() methods, Comparable and Comparator Interfaces, Thread Safety and Vector, Difference b/w Enumeration and Iterator, Type safety and Generics, Common algorithms and Collections class, Using Properties class for managing properties files;

Database Connectivity Using JDBC: Overview of native and ODBC Drives, Introduction to JDBC, Type of JDBC drivers, Usage of drivers, defining properties-based Connection Factory; Basic database operations: Insert, Delete, Update, and Select;

Prepared Statement: Statement, Prepared Statement, Setting Query parameters, Executing Queries;

Callable Statement: Creating PL/SQL Stored procedures and functions, Creating Callable statements, executing procedures & functions, Batch Updation, Transacting Queries, Programmatic initialization of database, ResultSetMetaData, DatabaseMetaData; Input/Output Stream, Stream Filters, Buffered Streams, Data input and Output Stream, Print Stream Random Access File,

Reflection: reflection API, newInstance()method, javap tool, creating javap tool, creating applet viewer, call private method, java 9 features.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

- CO1: Identify classes, objects, members of a class and relationships among them for a specific problem.
- CO2: Understand and demonstrate the concepts of garbage collection, polymorphism, inheritance etc.
- CO3: Do numeric (algebraic) and string-based computation.
- CO4: Understand and implement modularity as well as basic error-handling techniques.
- CO5: Develop, design and implement small multithreaded programs using Java language.
- CO6: Apply appropriate problem-solving strategies for the implementation of small/medium scale Java applications.

TEXT AND REFERENCE BOOKS:

1. E. Balaguruswamy, “Programming with Java”, TMH
2. Horstmann, “Computing Concepts with Java 2 Essentials”, John Wiley.
3. Decker &Hirshfield, “Programming.Java”, Vikas Publication.
4. Patrick Naughton and HerbertzSchidt, “Java-2 the complete Reference”, TMH
5. Sierra & bates, “Head First Java”, O’Reilly.

MICROPROCESSOR AND MICRO-CONTROLLER

Semester	IV				
Course code					
Category	Basic Science courses				
Course title	Microprocessor and Micro-controller				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To provide solid foundation on the fundamentals of microprocessors and applications, interfacing the external devices to the processor according to the user requirements thus, enabling to create novel products and solutions for real time problems.
2. Describe the architecture of 8086 microprocessors.
3. Develop programs for microprocessor and microcontrollers.
4. Compare microprocessors and microcontrollers.
5. Understand 8051, PIC and ARM microcontroller concepts, architecture and programming.

UNIT - I

THE 8086 MICRO-PROCESSOR

Introduction to 8086, Micro-processor architecture, Addressing modes, Instruction set and assembler directives, Assembly language programming, Modular Programming, Linking and Relocation, Stacks, Procedures, Macros, Interrupts and interrupt service routines, Byte and String Manipulation.

UNIT - II

8086 SYSTEM BUS STRUCTURE

8086 signals, Basic configurations, System bus timing, System design using 8086, I/O programming, Introduction to Multiprogramming, System Bus Structure, Multiprocessor configurations, Coprocessor, Closely coupled and loosely Coupled configurations, Introduction to advanced processors.

UNIT - III

I/O INTERFACING

Memory Interfacing and I/O interfacing, Parallel communication interface, Serial communication interface, D/A and A/D Interface, Timer, Keyboard /display controller, Interrupt controller, DMA controller, Programming and applications Case studies: Traffic Light control, LED display, LCD display, Keyboard display interface and Alarm Controller.

UNIT - IV

MICROCONTROLLER

Architecture of 8051, Special Function Registers(SFRs), I/O Pins Ports and Circuits, Instruction set, Addressing modes, Assembly language programming.

INTERFACING MICROCONTROLLER

Programming 8051 Timers, Serial Port Programming, Interrupts Programming, LCD & Keyboard Interfacing, ADC, DAC & Sensor Interfacing, External Memory Interface, Stepper Motor and Waveform generation.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Understand the operation and architecture of Intel 8085 microprocessor including Instruction Set Architecture, assembly language programming, timing and speed of operation.

CO2: Learn the operation of circuits for user interaction through switches, keyboard and display devices.

CO3: Understand the operation and architecture of Intel 8086 microprocessor including Instruction Set Architecture, assembly language programming, timing and speed of operation.

CO4: Understand the motivation and need for peripheral operations circuits for digital data exchange, timer, serial communication, merits of direct memory access, interrupt controller and other circuits.

CO5: Identify & diagnose common issues & errors that may arise during microprocessor & microcontroller programming.

CO6: Apply problem solving techniques to resolve hardware and software related problems.

TEXT AND REFERENCE BOOKS:

1. Microprocessors and interfacing: D V Hall; TMH
2. The 8088 & 8086 Microprocessors, Programming, interfacing, Hardware & Applications: Triebel & Singh; PHI
3. Microcomputer systems: the 8086/8088 Family: architecture, Programming &Design: Yu,Chang Liu & Glenn A Gibson; PHI.
4. Advanced Microprocessors and Interfacing: Badri Ram; TMH

DISCRETE MATHEMATICS

Semester	IV				
Course code					
Category	Basic Science courses				
Course title	Discrete Mathematics				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. To understand the basic theoretical concepts of set theory, functions, and relations.
2. To understand the basic theoretical concepts of logic systems and Boolean algebra.
3.
4. To express real-life problem of basic counting techniques and recurrence relations, algebraic structures.
5. The use of graphs theory concepts in real-life examples.

UNIT - I

SET THEORY, RELATIONS, FUNCTIONS, LOGIC AND PROPOSITIONAL CALCULUS

Set Theory: Introduction to set theory, Venn diagrams, Set operations, Algebra of sets, Duality, Finite and infinite sets, Counting principles, Power sets, Partitions, and Multi sets.

Relations: Cartesian product, Representation of relations, Types of relation, Binary relation, Equivalence relations, Partitions, Partial ordering relations, POSET, Hasse diagram, Lattices and its types.

Functions: Definition, Types of functions, Bijective functions, Composition of functions, Inverse functions, recursively defined functions, Finite and infinite sets, Countable and uncountable sets, Cantor's diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem.

Logic And Propositional Calculus: Introduction, Propositions and compound propositions, Logical operations, Propositions and truth tables, Tautologies, Contradictions, Logical equivalence, Algebra of propositions, Conditional and Bi-conditional statements, The use of Quantifiers.

UNIT - II

BASIC COUNTING TECHNIQUES AND RECURRENCE RELATION

Basic Counting Techniques: Pigeon-hole principle, Permutation and Combination, the Division algorithm: Prime Numbers, The GCD: Euclidean Algorithm, The Fundamental Theorem of Arithmetic.

Recursion And Recurrence Relation: Polynomials and their evaluation, Sequences, Introduction to AP, GP and AG Series, Partial Fractions, Recurrence Relation, Linear Recurrence Relations with

Constant Coefficients, Linear Homogeneous Recurrence Relations with Constant Coefficients, Particular Solution- Homogeneous Linear Difference Equations, Non-Homogeneous Linear Difference Equations, Total Solution, solving recurrence relation using generating functions.

UNIT - III

ALGEBRAIC STRUCTURES

Definitions and examples of Algebraic Structures with one Binary Operation: Semi Groups, Monoids, Groups, Semigroups, Subgroups, Abelian groups, Cosets, Normal Subgroup, Cyclic groups, Congruence Relation and Quotient Structures, Permutation Groups, Lagrange's Theorem, Homomorphism, Isomorphism, Automorphism.

Definitions and examples of Algebraic Structures with two Binary Operation: Rings, Integral Domain, Fields; Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form

UNIT - IV

GRAPHS THEORY: Introduction to graphs and their properties, Degree, Connectivity, Path, Cycle, Directed and undirected graphs, Subgraph, Bipartite Graphs, Regular Graphs, Connected Graphs, Multigraph and Weighted graph, Homomorphic and Isomorphic graphs, cut points and bridges, Paths and circuits, shortest path algorithm for weighted graphs, Eulerian paths and circuits, Hamiltonian path and circuits, Planar Graphs, Euler's formulae, Graph Colouring.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: To solve mathematical problems based on concepts of set theory, relations, functions and lattices.
CO2: To express logical sentences in terms of quantifiers and logical connectives.
CO3: To apply basic counting techniques to solve permutation and combination problems.
CO4: To solve recurrence relations.
CO5: To classify the algebraic structure of any given mathematical problem.
CO6: To evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.

TEXT AND REFERENCE BOOKS:

1. Kenneth H. Rosen, *Discrete Mathematics and its Applications*, 6th Edition, Tata McGraw Hill, 2011.
2. Satinder Bal Gupta: *A Text Book of Discrete Mathematics and Structures*, University Science Press, Delhi.
3. C. L. Liu and D. P. Mohapatra, *Elements of Discrete Mathematics A Computer Oriented Approach*, Tata McGraw Hill, 3rd Edition, 2008.
4. J.P. Trembley and R. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, Tata McGraw Hill – 13th reprint, 2012.
5. Richard Johnsonbaugh, *Discrete Mathematics*, 6th Edition, Pearson Education Asia, 2011.
6. S. Lipschutz and M. Lipson, *Discrete Mathematics*, Tata McGraw Hill, 3rd Edition, 2010.
7. B. Kolman, R. C. Busby and S. C. Ross, *Discrete Mathematical structures*, 6th Edition, PHI, 2010.

COMPUTER ORGANIZATION & ARCHITECTURE

Semester	IV				
Course code					
Category	Professional Core Courses				
Course title	Computer Organization & Architecture				
Scheme and Credits	L	T	P	Credits	
	3	0	0	3	
Classwork	30 Marks				
Exam	70 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OBJECTIVE:

1. How Computer Systems work & the basic principles.
2. Instruction Level Architecture and Instruction Execution.
3. The current state of art in memory system design.
4. How I/O devices are accessed and its principles.
5. To provide the knowledge on Instruction Level Parallelism.
6. To impart the knowledge on microprogramming.
7. Concepts of advanced pipelining techniques.

UNIT - I

Data representation: Data Types, Complements, Fixed-Point Representation, Conversion of Fractions, Floating-Point Representation, Gray codes, Decimal codes, Alphanumeric codes, Error Detection Codes.

Register Transfer and Microoperations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Microoperations, Logic Microoperations, Shift Microoperations, Arithmetic Logic Shift Unit.

UNIT - II

Basic Computer Organization and Design : Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instruction, Input-Output Instruction, Complete Computer Description, Design of Basic Computer, Design of Accumulator Logic.

Central Processing Unit : General Register Organization, Stack organization, Instruction Format, Addressing Modes, Data Transfer and Manipulation, Program Control, RISC, CISC.

UNIT - III

Pipelining: Parallel Processing, Amdahl's law, Pipelining, Arithmetic Pipeline, Instruction Pipeline, Pipeline Hazards, RISC Pipeline.

Parallel Processors: Introduction to Parallel Processors, Concurrent access to memory andCache Coherency.

Vector Processing: Vector Operations, Memory Interleaving, Supercomputers, Array Processors: Attached Array Processor, SIMD Array Processor.

UNIT - IV

Input-output Organization: I/O device interface, I/O transfers–program controlled, interrupt driven and DMA, Privileged and Non-Privileged Instructions, Software Interrupts.

Memory organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Associative Mapping, Direct Mapping, Set-Associative Mapping, Writing into Cache, Cache Initialization, Virtual Memory.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: outline the general concepts of digital electronics and computer organization and architecture.
CO2: discuss the basic components and their interfacing.
CO3: discuss the basic components and their interfacing.
CO4: analyse the effect of addressing modes on the execution time of a program.
CO5: analyse the effect of addressing modes on the execution time of a program.
CO6: Design of simple computer with different instruction sets.

TEXT AND REFERENCE BOOKS:

1. "Computer System Architecture", 3rd Edition by M.Morris Mano, Pearson.
2. "Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
3. "Computer Organization and Embedded Systems", 6th Edition by CarlHamacher, McGraw Hill Higher Education.
4. "Computer Architecture and Organization", 3rd Edition by John P. Hayes,WCB/McGraw-Hill
5. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
6. "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring andHarry F. Jordan, Pearson Education.

OPERATING SYSTEM LAB

Semester	IV				
Course code					
Category	Laboratory course				
Course title	Operating System Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

1. Introduction to UNIX File System.
2. File and Directory Related Commands in UNIX.
3. Essential UNIX Commands for working in UNIX environment.
4. I/O Redirection and Piping
5. Introduction to VI Editors.
6. Introduction of Processes in UNIX
7. Communication in UNIX and AWK.
8. Introduction of the concept of Shell Scripting.
9. Decision and Iterative Statements in Shell Scripting.
10. Writing the Shall Scripts for unknown problems.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: apply commands related to vi and Emacs editors, general utilities and file systems.

CO2: write basic shell scripts and use sed commands as well as awk programming.

CO3: analyse the results of memory management and disk management commands.

CO4: evaluate solutions for different operating system problems such as scheduling, memory management and file management.

CO5: create lab record for assignments that includes problem definitions, design of solutions and conclusions.

CO6: demonstrate use of ethical practices, self-learning and team spirit.

PROGRAMMING IN JAVA LAB

Semester	IV				
Course code					
Category	Laboratory course				
Course title	Programming in Java Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

1. Create a java program to implement stack and queue concept.
2. Write a java package to show dynamic polymorphism and interfaces.
3. Write a java program to show multithreaded producer and consumer application.
4. Create a customized exception and also make use of all the5 exception keywords.
5. Convert the content of a given file into the upper-case content of the same file.
6. Develop an analog clock using applet.
7. Develop a scientific calculator using swings.
8. Create an editor like MS-word using swings.
9. Create a servlet that uses Cookies to store the number of times a user has visited your servlet.
10. Create a simple java bean having bound and constrained properties.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: implement Java programs using object-oriented concepts for problem solving.
CO2: detect syntax and logical errors in java programs.
CO3: apply exception handling for making robust JAVA code.
CO4: design java applications using File I/O and GUI.
CO5: create lab record for assignments that includes problem definitions, design of solutions and conclusions.
CO6: Able to build dynamic user interfaces using applets and Event handling in java.

INTERNET OF THINGS LAB

Semester	IV				
Course code					
Category	Laboratory course				
Course title	Internet of Things Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

The student should have hands on experience in using various sensors like temperature, humidity, smoke, light, etc. and should be able to use control web camera, network, and relays connected to the Pi.

1. Start Raspberry Pi and try various Linux commands in command terminal window: ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, ping etc.
2. Run some python programs on Pi like:
   a) Read your name and print Hello message with name
   b) Read two numbers and print their sum, difference, product and division.
   c) Word and character count of a given string.
   d) Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.
3. Run some python programs on Pi like:
   a) Print a name 'n' times, where name and n are read from standard input, using for and while loops.
   b) Handle Divided by Zero Exception.
   c) Print current time for 10 times with an interval of 10 seconds.
   d) Read a file line by line and print the word count of each line.
4. a) Light an LED through Python program
   b) Get input from two switches and switch on corresponding LEDs
   c) Flash an LED at a given on time and off time cycle, where the two times are taken from a file.
5. a) Flash an LED based on cron output (acts as an alarm)
   b) Switch on a relay at a given time using cron, where the relay's contact terminals are connected to a load.
   c) Get the status of a bulb at a remote place (on the LAN) through web.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Understand the concept of Internet of Things.
CO2: Implement interfacing of various sensors with Arduino/Raspberry Pi.
CO3: Demonstrate the ability to transmit data wirelessly between different devices.
CO4: Show an ability to upload/download sensor data on cloud and server.
CO5: Examine various SQL queries from MySQL database

MICROPROCESSOR AND MICRO-CONTROLLER LAB

Semester	IV				
Course code					
Category	Laboratory course				
Course title	Microprocessor and Micro-controller Lab				
Scheme and Credits	L	T	P	Credits	
	0	0	2	1	
Classwork	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	02 Hours				

CONTENTS

1. Write a program using 8085 and verify for:
    a. Addition of two 8,bit numbers.
    b. Addition of two 8,bit numbers (with carry).
2. Write a program using 8085 and verify for:
    a. 8,bit subtraction (display borrow)
    b. 16,bit subtraction (display borrow)
3. Write a program using 8085 for multiplication of two 8, bit numbers by repeated addition method. Check for minimum number of additions and test for typical data.
4. Write a program using 8085 for multiplication of two 8, bit numbers by bit rotation method and verify.
5. Write a program using 8086 for finding the square root of a given number and verify.
6. Write a program using 8086 for copying 12 bytes of data from source to destination and verify.
7. Write a program using 8086 and verify for:
    a. Finding the largest number from an array.
    b. Finding the smallest number from an array.
8. Write a program using 8086 for arranging an array of numbers in descending order and verify.
9. Write a program using 8086 for arranging an array of numbers in ascending order and verify.
10. Write a program to interface a two,digit number using seven,segment LEDs. Use 8085/8086 microprocessor and 8255 PPI.
11. Write a program to control the operation of stepper motor using 8085/8086 microprocessor and 8255 PPI.
12. To study implementation & interfacing of Display devices Like LCD, LED Bar graph & seven segment display with Microcontroller 8051/AT89C51
13. To study implementation & interfacing of Different motors like stepper motor, DC motor & servo Motors.
14. Write an ALP for temperature & pressure measurement
15. Write a program to interface a graphical LCD with 89C51

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO6: Ability to analyze and understand the architecture, instruction set, and functioning of microprocessors and microcontrollers.

CO7: Proficiency in programming microprocessors and microcontrollers using assembly language and high level languages.

CO8: Skill in designing and implementing simple embedded systems by interfacing peripherals and devices with microprocessors and microcontrollers.

CO9: Competence in troubleshooting and debugging microprocessor and microcontroller based systems.

CO10: Understanding of the memory organization, input/output operations, and interrupt handling mechanisms in microprocessors and microcontrollers.

CO11: Familiarity with real-time operating systems and their applications in microcontroller, based systems.