Shri Vile Parle Kelavani Mandal's

Dwarkadas J. Sanghvi College of Engineering

(Autonomous College Affiliated to the University of Mumbai)

Scheme and Detailed Syllabus (DJS22)

Second Year B.Tech

in

INFORMATION TECHNOLOGY

(Semester III and IV)

Revision: 2 (2022)

With effect from the Academic Year: 2023-2024

1st July 2023

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Scheme for Second Year Undergraduate Program in Information Technology: Semester III (Autonomous)

(Academic Year 2023-2024)

SEMESTER III

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (A)						Continuous Assessment (B)					Aggregate (A+B)	Credits Earned	
			Theory (hrs)	Practical (hrs)	Tut (hrs)	Credits	Duration (hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total (A)	Term Test 1 (TT1)	Term Test 2 (TT2)	Total (TT1 & TT2)	Term Work Total	CA Total (B)			
1	DJS22ITC301	Discrete Structures	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	**3**	**4**
	DJS22ITT301	Discrete Structures Tutorial	--	--	1	**1**	--	--	--	--	--	--	--	--	--	25	**25**	**25**	**1**	
2	DJS22ITC302	Data Structures	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	**3**	**4**
	DJS22ITL302	Data Structures Laboratory	--	2	--	**1**	--	--		--	25	**25**	--	--	--	25	**25**	**50**	**1**	
3	DJS22ITC303	Database Management System	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	**3**	**4**
	DJS22ITL303	Database Management System Laboratory	--	2	--	**1**	--	--		--	25	**25**	--	--	--	25	**25**	**50**	**1**	
4	DJS22ITC304	Logic Design and Computer Architecture	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	**3**	**4**
	DJS22ITL304	Logic Design and Computer Architecture Laboratory	--	2	--	**1**	--	--	--	--	--	--	--	--	--	25	**25**	**25**	**1**	
5	DJS22ITC305	Operating System	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	**3**	**4**
	DJS22ITL305	Operating System Laboratory	-	2	--	**1**	--	--	25	--	--	**25**	--	--	--	25	**25**	**50**	**1**	
6	DJS22ITL306	Advanced Java Programming Laboratory	-	2	--	**1**	--	--	--	--	25	**25**	--	--	--	25	**25**	**50**	**1**	**1**
7	DJS22ITL307	Web Programming Laboratory	--	2	--	**1**	--	--	--	--	25	**25**	--	--	--	25	**25**	**50**	**1**	**1**
8	DJS22A2	Innovative Product Development	--	2	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
9	DJS22A3	Constitution of India	1	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
		Total	**16**	**14**	**1**	**22**	**10**	**325**	**25**	**0**	**100**	**450**	**100**	**75**	**175**	**175**	**350**	**800**	**22**	

Prepared by Checked by Head of the Department Vice Principal Principal

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Scheme for Second Year Undergraduate Program in Information Technology: Semester IV (Autonomous)

(Academic Year 2023-2024)

SEMESTER IV

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (A)						Continuous Assessment (B)					Aggregate (A+B)	Credits Earned	
			Theory (hrs)	Practical (hrs)	Tut (hrs)	Credits	Duration (hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total (A)	Term Test 1 (TT1)	Term Test 2 (TT2)	Total (TT1 & TT2)	Term Work Total	CA Total (B)			
1	DJS22ITC401	Applied Mathematics	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	3	**4**
	DJS22TT401	Applied Mathematics Tutorial	--	--	1	**1**	--	--	--	--	--	--	--	--	--	25	**25**	**25**	1	
2	DJS22ITC402	Formal Languages and Automata Theory	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	3	**4**
	DJS22ITL402	Formal Languages and Automata Theory Tutorial	--	--	1	**1**	--	--	--	--	--	--	--	--	--	25	**25**	**25**	1	
3	DJS22ITC403	Design and Analysis of Algorithms	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	3	**4**
	DJS22ITL403	Design and Analysis of Algorithms Laboratory	--	2	--	**1**	--	--	--	--	25	**25**	--	--	--	25	**25**	**50**	1	
4	DJS22ITC404	Computer Networks	3	--	--	**3**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	3	**4**
	DJS22ITL404	Computer Networks Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	25	**25**	**50**	1	
5	DJS22ITL405	Python Programing Laboratory	--	2	--	**1**	--	--	--	--	25	**25**	--	--	--	25	**25**	**50**	1	**1**
6	DJS22IHC1	Universal Human Values	2	--	--	**2**	2	65	--	--	--	**65**	20	15	35	--	**35**	**100**	2	**3**
7	DJS22IHT1	Universal Human Values Tutorial	--	--	1	**1**	--	--	--	--	--	--	--	--	--	25	**25**	**25**	1	
8	DJS22A4	Environmental Studies	1	--	--	**--**	--	--	--	--	--	--	--	--	--	--	--	--	--	**--**
9	DJS22A5	Innovative Product Development II	--	2	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
		Total	**15**	**08**	**3**	**20**	**10**	**325**	**25**	**0**	**50**	**400**	**100**	**75**	**175**	**150**	**325**	**725**	**20**	

Prepared by Checked by Head of the Department Vice Principal Principal

Continuous Assessment (A):

Course	Assessment Tools	Marks	Time (hrs.)
Theory	a. One Term test (based on 40 % syllabus)	20	1
	b. Second Term test (next 40 % syllabus) / presentation / assignment / course project / group discussion / any other.	15	1
	Total marks (a + b)	35	--
Audit course	Performance in the assignments / quiz / power point presentation / poster presentation / group project / any other tool.	--	As applicable
Laboratory	Performance in the laboratory and documentation.	25	
Tutorial	Performance in each tutorial & / assignment.	25	
Laboratory &Tutorial	Performance in the laboratory and tutorial.	50	

The final certification and acceptance of term work will be subject to satisfactory performance upon fulfilling minimum passing criteria in the term work / completion of audit course.

Semester End Assessment (B):

Course	Assessment Tools	Marks	Time(hrs.)
Theory / * Computer based	Written paper based on the entire syllabus.	65	2
	* Computer based assessment in the college premises.		
Oral	Questions based on the entire syllabus.	25	As applicable
Practical	Performance of the practical assigned during the examination and the output / results obtained.	25	2
Oral & Practical	Project based courses - Performance of the practical assigned during the examination and the output / results obtained. Based on the practical performed during the examination and on the entire syllabus.	As per the scheme	2

Prepared by Checked by Head of the Department Principal

Program: Information Technology **S.Y B.Tech** **Semester: III**
Course: Discrete Structures (DJS22ITC301)
Course: Discrete Structures Tutorial (DJS22ITT301)

Pre-requisite: Engineering Mathematics – I and Engineering Mathematics – II

Course Objectives: The objective of the course is to enable students to develop logical and mathematical thinking of the students. The course will also provide mathematical foundations required in further areas of computer science and Information technology.

Course Outcomes: On successful completion of this course, students should be able to:

1. Formulate and solve problems using set theory and use logical equivalence to understand and apply statements for solving problems.
2. Understand the concepts of relations and functions which is required for further topics.
3. Apply concepts of lattice and counting techniques to solve problems.
4. Apply the concept of groups for encoding and decoding binary data.
5.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Set Theory and Logic:** 1.1 **Sets:** Sets and elements, subsets, set operations, Venn diagram, power set, finite and infinite sets, cardinality of sets, principle of inclusion and exclusion, cartesian product of sets. 1.2 **Logic:** Propositional logic, propositional equivalence, truth tables, converse-inverse-contrapositive of conditional statement, universal and existential quantifiers.	**08**
2	**Relations:** Definition, domain and range of a relation, pictorial representation (Digraph and matrix) of a relation, types of relation, equivalence relation, closure of relations, Warshall's algorithm to compute transitive closure, partially ordered set (POSET).	**05**
3	**Lattice and Functions** **Lattice:** Hasse diagram, least upper bound (LUB) and greatest lower bound (GLB), definition of Lattice, complemented and distributive Lattice, topological sorting. **Functions:** Definition, domain and range of functions, types of functions: one-one, onto, bijective, composition of functions, inverse of a function.	**07**
4	**Techniques of Counting:** Basics of counting, sum and product rule, permutations and combinations, pigeonhole principle, Recurrence relations, linear recurrence relations with constant coefficients, Solving second-order homogeneous linear recurrence relation.	**05**
5	**Algebraic System:** **Groups:** Semi-groups, monoids, groups, abelian groups, subgroups, permutation groups, cyclic groups, cosets, normal groups, Lagrange's theorem, homomorphism, isomorphism, automorphism. **Rings:** Rings, subrings, integral domains, ring homomorphism, ring isomorphism, fields.	**09**
6	**Coding Theory:** Encoding function, codewords, Hamming distance, parity check code, error detection, minimum distance of an encoding function, group codes, Decoding function, error correction, maximum likelihood technique.	**05**

List of Tutorials:

1. Set theory
2. Logic
3. Relations

4. Functions
5. Lattice
6. Counting techniques
7. Groups
8. Isomorphisms
9. Rings
10. Coding

Minimum eight tutorials based on syllabus will be conducted. Mini project relevant to the subject may be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text books:

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", 8th Edition, Mc Graw Hill, 2021.
2. Seymour Lipschutz and Marc Lars Lipson, Adapted by Varsha H. Patil, "Discrete Mathematics", 3rd Edition, Schaum's Outlines., 2017.

Reference Books:

1. Bernard Kolman, Robert Busby, Sharon C. Ross, "Discrete Mathematical Structures", 6th Edition, Pearson, 2015.
2. C. L. Liu, D P Mohapatra, "Elements of Discrete Mathematics (A computer oriented approach)", 4th Edition, Mc Graw Hill, 2017.
3. Susanna S. Epp, "Discrete Mathematics with Applications", 5th Edition, CENGAGE, 2019.
4. Ralph P. Grimaldi, "Discrete and combinatorial Mathematics (An applied introduction)", 5th Edition, Pearson (Addison Wesley), 2004.

Program: Information Technology **S.Y B.Tech** **Semester: III**
Course: Data Structures (DJS22ITC302)
Course: Data Structures Laboratory (DJS22ITL302)

Pre-requisite: Knowledge of C Programming

Course Objectives: The objective of the course is to introduce and familiarize students with linear and non-linear data structures, their use in fundamental algorithms and design & implementation of these data structures. To introduce students to the basics of algorithms and time complexity. To familiarize students to various sorting and searching techniques, and their performance comparison.

Course Outcomes: On successful completion of this course, student should be able to:
1. Understand the concept of time and space complexity for algorithms.
2. Assimilate the concept of various linear and non-linear data structures.
3. Solve the problem using appropriate data structure.
4. Implement appropriate searching and sorting technique for a given problem.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Basics of Algorithms:** Algorithms, Characteristics of an Algorithm, Time and Space Complexities, Order of Growth functions, Preliminary Asymptotic Notations. **Data Structures:** Introduction, Need of Data Structures, Types of Data Structures, Abstract Data Types (ADT).	**04**
2	**Linear Data Structures – LIST:** List as an ADT, Array-based implementation, Linked List implementation, Singly linked lists, Circularly linked lists, Doubly-linked lists, All operations (Insertion, Deletion, Merge, Traversal, etc.) and their analysis, Applications of linked lists - (Polynomial Addition).	**06**
3	**Linear Data Structure – STACK:** Stack as an ADT, Operations, Array and Linked List representation of Stack, Applications – Reversing data, Conversion of Infix to prefix and postfix expression, Evaluation of postfix and prefix expressions, Balanced parenthesis, etc. **Linear Data Structure – QUEUE:** Queue as an ADT, Operations, Implementation of Linear Queue, Circular and Priority Queue using arrays and Linked List, DEQueue, Applications – Queue Simulation.	**07**
4	**Non-Linear Data Structure – TREES:** Tree Terminologies, Tree as an ADT, Binary Tree - Operations, Tree Traversals, Binary Search Tree (BST) - Operations, Expression Trees **Height Balanced Tree**: Creation of AVL Tree, **Heap**- Operations on heap, **Applications** - Huffman coding	**08**
5	**Non-Linear Data Structure – GRAPHS:** Graph Terminologies, Types of Graphs, Representation of Graph using arrays and Linked List, Breadth-First Search (BFS), Depth–First Search (DFS), **Applications of Graphs** -Topological sorting.	**06**
6	**Searching**- Linear Search, Binary Search and Fibonacci search. **Sorting:** Bubble Sort, Selection Sort, Heap Sort, Insertion Sort, Radix Sort, Merge Sort, Quick Sort. Analysis of Searching and Sorting Techniques. **Hashing:** Hash Functions, Overflow handling, Collision & Collision Resolution Techniques, Linear hashing, Hashing with chaining, Separate Chaining, Open Addressing, Rehashing and Extendible hashing.	**08**

List of Laboratory Experiments: (Any 10 to 12)

1. Implementation of Linked List using menu driven approach.
2. Implementation of different operations on linked list –copy, concatenate, split, reverse, count no. of nodes etc.
3. Implementation of polynomials operations (addition, subtraction) using Linked List.
4. Implementation of stack using menu driven approach.
5. Implementation of Infix to Postfix conversion.
6. Implementation of prefix and postfix evaluation using menu driven approach.
7. Implementation of parenthesis checker using stack.
8. Implementation of Linear queue using menu driven approach.
9. Implementation of circular queue using menu driven approach.
10. Implementation of double ended queue using menu driven approach.
11. Implementation of Priority queue program using array and Linked list.
12. Implementation of Binary Tree using menu driven approach.
13. Implementation of Binary Tree Traversal.
14. Implementation of BST.
15. Implementation of various operations on tree like – copying tree, mirroring a tree, counting the number of nodes in the tree, counting only leaf nodes in the tree.
16. Implementation of Graph traversal using menu driven program (DFS & BSF).
17. Implementation of Selection sort, Radix sort using menu driven.
18. Implementation of Heap & Heap Sort using menu driven program.
19. Implementation of Advanced Bubble Sort and Insertion Sort using menu driven Program.
20. Implementation of searching methods (Index Sequential, Fibonacci search, Binary Search) using menu driven program.
21. Implementation of hashing functions with different collision resolution technique

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

Books Recommended:

Text books:

1. R. F. Gilberg and B. A. Forouzan, "Data Structures – A Pseudocode Approach with C", 2nd Edition, Cengage Learning, 2005.
2. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, "Fundamentals of Data Structures in C", 2nd Edition, W. H. Freeman and Company, 2008.

Reference Books:

1. Mark A. Weiss, "Data Structures and Algorithm Analysis in C", 4th Edition, Pearson, 2014.
2. M. T. Goodritch, R. Tamassia and D. Mount, "Data Structures and Algorithms in C++", 2nd Edition, Wiley, 2011.
3. Kruse, Leung and Tondo, "Data Structures and Program Design in C", 2nd Edition, Pearson Education, 2013.
4. Tenenbaum, Langsam and Augenstein, "Data Structures using C", 2nd Edition, Pearson, 2015.
5. Reema Thareja, "Data Structures using C", 2nd Edition, Oxford University Press, 2014.
6. Seymour Lipschutz, "Data Structures", Schaum's Outline Series, 1st Edition, Tata McGraw-Hill, 2014.

Prepared by Checked by Head of the Department Vice Principal Principal

Program: Information Technology **S.Y B.Tech** **Semester: III**
Course: Database Management System (DJS22ITC303)
Course: Database Management System Laboratory (DJS22ITL303)

Pre-requisite: Computer Basics

Course Objectives: The course intends to introduce the students to the management of database systems, with an emphasis on how to design, organize, maintain and retrieve information efficiently and effectively from a database.

Course Outcomes: On successful completion of this course, student should be able to:
1. Design an optimized database.
2. Construct SQL queries to perform operations on the database.
3. Demonstrate appropriate transaction management and recovery techniques for a given problem.
4. Apply indexing mechanisms for efficient retrieval of information from database.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction Database Concepts:** Introduction, Characteristics of databases, File system v/s Database system, Users of Database system, Schema and Instance, Data Independence, DBMS system architecture, Database Administrator.	03
2	**Relational Data Model:** **Entity–Relationship Model:** The Entity-Relationship (ER) Model: Entity types: Weak and strong entity sets, Entity sets, Types of Attributes, Keys, Relationship constraints: Cardinality and Participation, Extended Entity-Relationship (EER) Model: Generalization, Specialization and Aggregation **Relational Model:** Introduction to the Relational Model, relational schema and concept of keys, Mapping the ER and EER Model to the Relational Model, Introduction to Object-Relational Databases, ORDBMS Vs Relational Databases **Relational Algebra**: Unary and Set operations, Relational Algebra Queries.	09
3	**Structured Query Language (SQL):** Overview of SQL, Data Definition Commands, Data Manipulation commands, Integrity constraints - key constraints, Domain Constraints, Referential integrity, check constraints, Data Control commands, Transaction Control Commands, Set and String operations, aggregate function - group by, having, Views in SQL, joins, Nested and complex queries, Triggers, Security and authorization in SQL	09
4	**Relational–Database Design:** Pitfalls in Relational-Database designs, Concept of normalization, Function Dependencies, Normal Forms- 1NF, 2NF, 3NF, BCNF	05
5	**Transaction Management and Recovery:** Transaction Concept, ACID properties, Transaction States, Implementation of atomicity and durability, Concurrent Executions, Serializability, Concurrency Control Protocols: Lock-based, Timestamp based, Validation Based, Deadlock Handling, Recovery System: Failure classification, Log based recovery, Shadow Paging, ARIES recovery algorithm.	09
6	**Indexing Mechanism:** Hashing techniques, Types of Indexes: Single Level Ordered Indexes, Multilevel Indexes, Overview of B-Trees and B+ Trees.	04

List of Laboratory Experiments:
1. To draw an ER diagram for a problem statement.
2. Map the ER/EER to relational schema.
3. To implement DDL and DML commands with integrity constraints.
4. To access & modify Data using basic SQL.
5. To implement Joins and Views.

6. To implement Subqueries.
7. To implement triggers.
8. Examine the consistency of database using concurrency control technique (Locks)
9. To simulate ARIES recovery algorithm.
10. To implement B-trees/B+ trees.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

Books Recommended:

Text books:

1. Korth, Silberschatz, Sudarshan, "Database System Concepts", 7th Edition, McGraw – Hill, 2019.
2. Elmasri and Navathe, "Fundamentals of Database Systems", 7th Edition, Pearson Education, 2021.
3. G Peter Rob and Carlos Coronel, "Database Systems Design, Implementation and Management", 5th Revised Edition, Thomson, 2002
4. G. K. Gupta, "Database Management Systems", 3rd Edition, McGraw – Hill, 2018

Reference Books:

1. Dr. P.S. Deshpande, "SQL and PL/SQL for Oracle 10g, Black Book", Dreamtech Press, 2012
2. Sharanam Shah, Vaishali Shah, "Oracle for Professional", 1st Edition, Shroff Publishers & Distributers Private Limited, 2008
3. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", 3rd Edition, McGraw – Hill, 2014.
4. Patrick Dalton, "Microsoft SQL Server Black Book", 11th Edition, Coriolis Group, U.S., 1997
5. Lynn Beighley, "Head First SQL", 1st Edition, O'Reilly Media, (28 August 2007)

Program: Information Technology **S.Y B.Tech Semester: III**
Course: Logic Design and Computer Architecture (DJS22ITC304)
Course: Logic Design and Computer Architecture Laboratory (DJS22ITL304)

Pre-requisite: Knowledge of Basic Electrical Engineering.

Course Objectives: The aim of the course is to provide a comprehensive introduction to digital circuit design. The course familiarizes students to number system representations, binary codes, binary arithmetic and Boolean algebra, its axioms and theorems, and its relevance to digital logic design. The student will analyze and design combinational and sequential circuits for implementing various digital circuits which helps them to comprehend the computer architecture and have a thorough understanding of the basic structure and operation of a digital computer.

Course Outcomes: On successful completion of this course, student should be able to:
1. Design combinational circuits.
2. Design sequential circuits.
3. Understand the architecture and functionality of central processing unit.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Binary Arithmetic, Codes, and Boolean Algebra:** Binary number systems, signed binary numbers, Binary arithmetic – addition, subtraction using 1's and 2's complements, multiplication, and division. Gray Code, BCD Code, and Excess-3 Code. Basic Gates (NOT, AND, OR), Universal Gates (NAND, NOR); Boolean algebra theorems and properties, Canonical form, Standard SOP and POS form, Reduction of Boolean functions using algebraic method, K -map method (2, 3, 4 Variable), Quine Mc-Cluskey minimization technique.	**07**
2	**Combinational Circuits:** Introduction to combinational circuits, half adder and full adder, half subtractor and full subtractor, n-bit parallel adder and subtractor, look ahead – carry adder, Multiplexer and De-Multiplexer, Multiplexer and De-Multiplexer tree, Encoder and Decoder.	**06**
3	**Sequential circuits:** Introduction to sequential circuits, Flip Flops: SR, JK, D, T, master slave flip flop, truth table, excitation table and conversion, Registers: Shift register, SISO, SIPO, PISO, PIPO, Bi-Directional and Universal shift register, Counters: Synchronous and Asynchronous counters, Mod counter, Up-Down counter, Ring Counter, Twisted Ring Counter.	**09**
4	**Computer Evolution and Performance:** History of Computers - First, Second, Third, and Later generations; Organization and Architecture, Structure and Function, Designing for Performance, Evolution of Intel x86 Architecture, Performance Assessment - Clock Speed, Instructions per second, Amdahl's Law.	**04**
5	**Top-Level View of Computer Function and Interconnection:** Computer Function – Instruction Fetch and Execute, Interrupts, I/O Function; Interconnection Structures, Bus Interconnection, Cache Memory, Internal Memory, External Memory, Input/Output – I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access.	**06**
6	**The Central Processing Unit:** The ALU, Integer Representation, Integer Arithmetic, Floating-Point Representation, Floating-Point Arithmetic, Machine Instruction Characteristics, Types of Operations – Data Transfer, Arithmetic, Logical, Conversion, Input/Output; Addressing Modes – Immediate, Direct, Indirect, Register, Register Indirect; Register Organization, Instruction Cycle.	**07**

List of Laboratory Experiments: (Any 10 to 12)

1. To study logic gates & universal gates and verify their truth tables.
2. To verify the logical expression using truth table.
    a. To implement basic gates using universal gates.
3. Implement full adder and full subtractor using logic gates and verify their truth tables.
4. Implement the functions using 8:1 Multiplexer.
    a. Implement 1:8 De-multiplexer.
5. Implement JK Flip Flop using only NAND gates and verify its truth table.
6. Implement S-R Flip Flop and verify its truth table.
7. To study and verify the following functions of Universal Shift Register:
    a. Parallel loading
    b. Right shift
    c. Left shift
8. Implement Synchronous and Asynchronous counter.
9. Implement Direct, Associative Mapping Technique in Cache Allocation.
10. Implement Set Associative Mapping Technique in Cache Allocation.
11. Implement Integer and Floating-point arithmetic operations.
12. Implement Addressing Modes.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

Books Recommended:

Text books:

1. R. P. Jain, "Modern Digital Electronics", 4th Edition, Tata McGraw Hill, 2009.
2. M. Morris Mano, "Digital Logic and computer Design", 1st Edition, Pearson Education India, 2016.
3. Balbaniam, Carlson, "Digital Logic Design Principles", 1st Edition, John Wiley & Sons Inc, 2001.
4. William Stallings, "Computer Organization and Architecture: Designing for Performance", 10th Edition, Pearson, 2015.

Reference Books:

1. Anand Kumar, "Fundamentals of Digital Circuits", 4th Edition, Prentice Hall India, 2003.
2. Donald P Leach, Albert Paul Malvino, "Digital Principles and Applications", 8th Edition, Tata McGraw Hill, 2014.
3. John Hennessy, David Patterson, "Computer Architecture: A Quantitative Approach", 6th Edition, Morgan Kaufmann, 2017.

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: Information Technology **S.Y B. Tech** **Semester: III**

Course: Operating System (DJS22ITC305)

Course: Operating System Laboratory (DJS22ITL305)

Pre-requisite: Knowledge of

1. Programming Language C.
2. Basics of Hardware, i.e., ALU, RAM, ROM, HDD, etc.

Course Objectives: The objective of this course is to familiarize students with the functionality of an Operating System, its basic components & interaction among them. The course will also expose students to analyze and evaluate different policies for scheduling, deadlocks, memory management, synchronization, file management & I/O and implement these policies using a suitable programming language.

Course Outcomes: On successful completion of this course, student should be able to:

1. Understand the role of Operating System in terms of process, memory, file and I/O management.
2. Apply appropriate process scheduling, memory mapping and disk scheduling methods.
3. Identify the need of concurrency and apply the appropriate method to solve the concurrency or deadlock problem.
4. Apply and analyze different techniques of file and I/O management.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Operating System:** Operating System Objectives, basic functions and services, Evolution of operating system, Operating System structures (monolithic, microkernel), Types of Operating Systems: Batch, multiprogramming. Multitasking, time sharing, parallel, distributed & real-time O.S., Linux OS, Mobile OS, System calls.	**04**
2	**Process Management:** Concept of a Process, Process States, Process Description, Process Control Block, Operations on Processes. **Threads**: Definition and Types, Concept of Multithreading, **Scheduling:** Types of Scheduling: Preemptive and, Non-preemptive, Scheduling algorithms and their performance evaluation: FCFS, SJF, SRTF, Priority based, Round Robin.	**07**
3	**Process Synchronization and Deadlocks** **Concurrency:** Principles of Concurrency, Inter-Process Communication, Process/Thread Synchronization. **Mutual Exclusion:** Requirements, Hardware and Software Support, Semaphores and Mutex, Monitors, Classical synchronization problems: Producer and Consumer problem, Readers/Writers Problem.	**07**
4	**Deadlock:** Principles of deadlock, Conditions for deadlock, Resource Allocation Graph, Deadlock Prevention, Deadlock Avoidance: Banker's Algorithm for Single & Multiple Resources, Deadlock Detection and Recovery. Dining Philosophers Problem.	**07**
5	**Memory Management:** Memory Management Requirements, Memory Partitioning: Fixed Partitioning, Dynamic Partitioning, Memory Allocation Strategies: Best-Fit, First Fit, Worst Fit, Next Fit, Relocation, Paging, Segmentation. Virtual Memory: Demand Paging, Structure of Page Tables, Page Replacement Strategies: FIFO, Optimal, LRU, LFU, Thrashing.	**07**
6	**File System and I/O Management** **File Management:** Overview, File Organization and Access, Secondary Storage Management: File Allocation Methods **Input /Output Management:** I/O Management and Disk Scheduling: I/O Devices, I/O Buffering, Disk Scheduling algorithm: FCFS, SSTF, SCAN, CSCAN, LOOK, C-LOOK. RAID	**07**

List of Laboratory Experiments:

1. Explore the internal commands of linux and Write shell scripts to do the following: Display top 10 processes in descending order Display processes with highest memory usage. Display current logged in user and logname. Display current shell, home directory, operating system type, current path setting, current working directory. Display OS version, release number, kernel version. Illustrate the use of sort, grep, awk, etc.
2. System calls for file manipulation.
3. CPU scheduling algorithms like FCFS, SJF, Round Robin etc.
4. There is a service counter which has a limited waiting queue outside it. It works as follows:
    a. The counter remains open till the waiting queue is not empty.
    b. If the queue is already full, the new customer simply leaves.
    c. If the queue becomes empty, the outlet doors will be closed (service personnel sleep).

    Whenever a customer arrives at the closed outlet, he/she needs to wake the person at the counter with a wake-up call. Implement the above described problem using semaphores or mutexes along with threads. Also show how it works, if there are 2 service personnel, and a single queue. Try to simulate all possible events that can take place, in the above scenario.
5. Implement Banker's Algorithm for deadlock avoidance.
6. Implement Placement algorithms (Best, First, Worst fit).
7. Implement various page replacement policies (LRU, FIFO, Optimal).
8. Implement File allocation techniques (Sequential, Indexed, Linked).
9. Implement disk scheduling algorithm FCFS, SSTF, SCAN, CSCAN etc.
10. Using the CPU-OS simulator analyze and synthesize the following:
    a. Process Scheduling algorithms.
    b. Thread creation and synchronization.
    c. Deadlock prevention and avoidance.
11. Building a scheduler in XV6.
12. Building own file system.
13. Building a device driver.
14. Implementing a kernel routine in Linux Operating System.

Books Recommended:

Textbooks:

1. Abraham Silberschatz, Greg Gagne, Peter Baer Galvin, "Operating System Concepts", 8th Edition, Wiley,2018.
2. Tanenbaum, "Modern Operating System", 4th Edition, Pearson Education, 2014.
3. William Stallings, "Operating Systems: Internal and Design Principles", 8th Edition, Pearson, 2014.
4. Randal. K. Michael, "Mastering Shell Scripting", 2nd Edition, Wiley Publication, 2008.

Reference Books:

1. A Tanenbaum, "Operating System Design and Implementation", 3rd Edition, Pearson, January 2015.
2. Phillip A. Laplante, Seppo J. Ovaska, "Real Time Systems Design and Analysis", 4th Edition, Wiley-IEEE Press,Dec 2011.
3. Naresh Chauhan, "Principles of Operating Systems", 1st Edition, Oxford University Press, 2014.

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: Information Technology **S.Y B. Tech** **Semester: III**
Course: Advanced Java Programming Laboratory (DJS22ITL306)

Pre-requisite: Structured programming using C, Object Oriented Programming using Java.

Course Objectives: The objective of the course is to introduce and familiarize students with advanced concepts that go beyond Core Java – most importantly the APIs defined in Java Enterprise Edition. To expose students to a set of services, API and protocols, that allows the functionality which is a requisite for developing a multi-tiered and enterprise-level application. To familiarize students with Advanced Java framework like Spring, that allows us to develop secure transaction-based applications.

Course Outcomes: On successful completion of this course, student should be able to:

1. Modify the behavior of methods, classes, and interfaces at runtime.
2. Develop enterprise applications.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Java Collections:** Collections in Java, basic data structures, arrays and lists, stacks, and queues, sets and maps.	02
2	**Generics:** Basic generics, bounded type parameters, type inference, wildcards, type erasure.	04
3	**Java Beans:** Java Beans Introduction to Java Beans, Java Beans API, class Beans, JAR Files, Building Java Beans with NETBEAN IDE, Source code Generated by IDE, Java Beans Project: Construct a Bean Containing a Label and scroll bar, illustrate mouse, illustrate working of radio buttons, combo box.	04
4	**JDBC:** Introduction, JDBC Architecture, installing MySQL and MySQL Connector/J, JDBC Environment, Setup, Establishing JDBC Database connection, ResultSet Interface, Creating JDBC Application, JDBC Batch Processing, JDBC Transaction Management, Application Programs.	04
5	**Java Reflection API:** Modifiers and Security, Accessing Fields, Accessing Methods, Accessing Constructors, What About Arrays? Accessing Generic Type Information, Accessing Annotation Data, Dynamic Interface Adapters.	04
6	**The Stream API:** Stream Basics, Stream Interfaces, How to Obtain a Stream, A Simple Stream Example, Reduction Operations, Using Parallel Streams, Mapping, Collecting, Iterators and Streams, Use an Iterator with a Stream, Use Spliterator.	04
7	**Introduction to Spring:** Spring Architecture, Introduction to all modules of Spring, Setting up spring.	04

List of Laboratory Experiments:

1. Implementation of different collection types (stacks, queues, vectors, etc.)
2. Creation of generic classes, methods
3. Creating JAR files
4. Building Java Beans with NETBEAN IDE
5. Construct a Bean Containing label, scroll bar, radio button and combo box.
6. Creating JDBC application
7. Use reflection API to examine or modify the behaviour of methods, classes, and interfaces at runtime.
8. Using streams API to implement program logic by composing functions and executing them in a data flow.
9. Use Spring framework to create enterprise grade applications.

Books Recommended:

Textbooks:

1. Anita Seth, B.L. Juneja, "JAVA: ONE STEP AHEAD", Oxford University Press; 1st Edition, 2017.
2. Patrick Niemeyer, Daniel Leuck, "Learning Java", O'Reilly Media, 4th Edition, 2013.

Reference Books:

1. Herbert Schildt, "Java: The Complete Reference", 9th Edition, O'Reilly for Higher Education (Firm), Oracle Press, 2014
2. Craig Walls, "Spring in Action", 6th Edition, Manning Publications
3. Uttam K. Roy, "Advanced Java Programming", Oxford University Press, 2015.
4. D.T. Editorial Services, "Java 8 Programming Black Book", 8th Edition, Dreamtech Press, 2015.
5. Cay S. Horstmann, Gary Cornell, "Core Java™ 2: Volume II–Advanced Features", 9th Edition Prentice Hall PTR, 2012.

Program: Information Technology **S.Y B. Tech** **Semester: III**
Course: Web Programming Laboratory (DJS22ITL307)

Pre-requisite: Knowledge of how Internet works

Course Objective: As the internet plays an important role in our daily life, it becomes essential to understand the significance of various protocols, methods and means that are used for website development. This course aims to provide students with an introduction to client-side scripting and dynamic Web application development. The students will acquire knowledge and skills for creation of a responsive website by developing interactive user interfaces with data validation.

Course Outcomes: On successful completion of this course, student should be able to:
1. Develop web applications.
2. Work effectively as a member of a team.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Static web page design – HTML and HTML5:** What is HTML? What is an HTML Element? HTML Page Structure Formatting Fonts and Headings Basic Tags (Div, Para, Anchor Tags, Hyperlinks, buttons), List, Background, Media Elements (audio, video, images), Table, Form, Semantic Elements **CSS and CSS3:** CSS Introduction, Ways of inserting a style sheet, Selectors (id, class), Backgrounds, Colour, Font, Borders, Margins, Padding, Height, Weight, Display, Overflow, Float, Opacity, Position, Pseudo Selectors. Responsive web design using Media Queries - Supporting Different Viewports, Embracing Fluid Layout	**04**
2	**Introduction to JavaScript: Basic** Syntax, Inserting JavaScript in HTML, Variables and their scope, Data types, operators, strings. **Arrays & Loops:** Arrays, objects and methods For, if else, while, break, continue, JSON and methods. **Client-side scripting – JavaScript:** **Functions & Error handling:** Functions, Arrow functions, Break, continue, Map, Filter**,** Reduce, Error handling, Asynchronous Java Script and Event loop. **DOM:** Browser objects and DOM objects, Event handling, Form validation, Cookies.	**04**
3	**Introduction to React JS:** Introduction to React, Adding React to HTML page, Introducing JSX, Rendering elements Introduction to DOM, CSS Syntax, **Components:** Components, Props, Events	**04**
4	**React JS: Hooks:** useState, useEffect, **Conditional Rendering Mapping:** Lists and Keys**.** **React-router-dom:** BrowserRouter, Routes, Route**,** Link and useNavigation	**04**
5	**Server-Side Scripting:** NodeJS Modules, NodeJS HTTP Modules, NodeJS file systems, NodeJS URL Module, NodeJS Events, NodeJS Functions	**04**
6	**MongoDB:** Understanding MongoDB, MongoDB Data Types, Administering User Accounts, Configuring Access Control, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Accessing and Manipulating Databases, Manipulating MongoDB Documents from Node.js, Accessing MongoDB from Node.js, Using Mongoose for Structured Schema and Validation. **REST API:** Examining the rules of REST APIs, Evaluating API patterns, Handling typical CRUD functions (create, read, update, delete), Using Express and Mongoose to interact with MongoDB, Testing API endpoints.	**06**

List of Laboratory Experiments:

1. HTML: Design a website using only HTML
2. Style elements of HTML website using CSS.
3. Make the website responsive using Bootstrap.
4. Design a website using TailWind CSS.
5. Programs based on objects in JavaScript.
6. Develop and demonstrate JavaScript with POP-UP boxes and functions.
7. Form Validation using JavaScript.
8. Event Loop using JavaScript Visualizer.
9. Write a program to create React Components.
10. Write a program to add Bootstrap to React.
11. Write a program in React JS for Event Handling.
12. Installation and Configuration of Node.js server
13. Program based on inbuilt functions in Node.js
14. Create REST Web services using MongoDB
15. Mini-Project based on React JS.

Books Recommended:

Text books:

1. DT Editorial Services, "HTML5 Black Book", 2nd Edition, Dreamtech Press, 2016.
2. Ben Frain, "Responsive Web Design with HTML5 and CSS3", 2nd Edition, Packt Publishing, 2015.
3. Steve Suehring, "JavaScript Step by Step", 3rd Edition, Pearson Education, 2013.
4. Stoyan Stefanov, "React Up Running Building Web Applications", 1st Edition, O'Reilly Media Inc., 2016.
5. David Sklar, "Learning PHP 5", 1st Edition, O'Reilly Media Inc., 2004.

Reference Books:

1. Benjamin LaGrone, "HTML5 and CSS3 Responsive Web Design Cookbook", 1st Edition, Packt Publishing, 2013.
2. DT Editorial Services, "Web Technologies: Black Book", 1st Edition, Dreamtech Press, 2018.
3. Christopher Schmitt, Kyle Simpson, "HTML5 Cookbook", 1st Edition, O'Reilly Media Inc., 2011.
4. Uttam K. Roy, "Web Technologies", 1st Edition, Oxford University Press, 2010.
5. Greg Sidelnikov, "React. Js Book: Learning React JavaScript Library from Scratch", 1st Edition, IndependentlyPublished, 2017.
6. Luke Welling; Laura Thomson, "PHP and MySQL Web Development", 5th Edition, Addison-Wesley ProfessionalPTG, 2017.

Program: Information Technology **S.Y B. Tech** **Semester: III**
Course: Innovative Product Development I (DJS22A2)

Course Objectives:

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuring that a solution is found out to address the same by designing and developing an innovative product.
2. To familiarize the students with the process of designing and developing a product, while they work as part ofa team.
3. To acquaint the students with the process of applying basic engineering fundamentals, so as to attempt at thedesign and development of a successful value added product.
4. To inculcate the basic concepts of entrepreneurship and the process of self-learning and research required to conceptualize and create a successful product.

Course Outcomes: On successful completion of this course, student should be able to:

1. Identify the requirement for a product based on societal/research needs.
2. Apply knowledge and skills required to solve a societal need by conceptualizing a product, especially while working in a team.
3. Use standard norms of engineering concepts/practices in the design and development of an innovative product.
4. Draw proper inferences through theoretical/ experimental/simulations and analyze the impact of the proposed method of design and development of the product.
5. Develop interpersonal skills, while working as a member of the team or as the leader.
6. Demonstrate capabilities of self-learning as part of the team, leading to life-long learning, which could eventually prepare them to be successful entrepreneurs.
7. Demonstrate product/project management principles during the design and development work and also excel in written (Technical paper preparation) as well as oral communication.

Guidelines for the proposed product design and development:

- Students shall form a team of 3 to 4 students (max allowed: 5-6 in extraordinary cases, subject to the approvalof the department review committee and the Head of the department).
- Students should carry out a survey and identify the need, which shall be converted into conceptualization of a product, in consultation with the faculty supervisor/head of department/internal committee of faculty members.
- Students in the team shall understand the effective need for product development and accordingly select the best possible design in consultation with the faculty supervisor.
- Students shall convert the best design solution into a working model, using various components drawn fromtheir domain as well as related interdisciplinary areas.
- Faculty supervisor may provide inputs to students during the entire span of the activity, spread over 2 semesters, wherein the main focus shall be on self-learning.
- A record in the form of an activity log-book is to be prepared by each team, wherein the team can record weekly progress of work. The guide/supervisor should verify the recorded notes/comments and approve the same on a weekly basis.
- The design solution is to be validated with proper justification and the report is to be compiled in a standard format and submitted to the department. Efforts are to be made by the students to try and publish a technical paper, either in the institute journal, “Techno Focus: Journal for Budding Engineers” or at a suitable publication,approved by the department research committee/ Head of the department.
- The focus should be on self-learning, capability to design and innovate new products as well as on developing the ability to address societal problems. Advancement of entrepreneurial capabilities and quality development of the students through the year long course should ensure that the design and development of a product of appropriate level and quality is carried out, spread over two semesters, ie during the semesters III and IV.

Guidelines for Assessment of the work:

- The review/ progress monitoring committee shall be constituted by the Head of the Department. The progressof

design and development of the product is to be evaluated on a continuous basis, holding a minimum of two reviews in each semester.

- In the continuous assessment, focus shall also be on each individual student's contribution to the team activity,their understanding and involvement as well as responses to the questions being raised at all points in time.
- Distribution of marks individually for the both reviews as well as for the first review during the subsequent semester shall be as given below:
    - Marks awarded by the supervisor based on log-book 20
    - Marks awarded by review committee 20
    - Quality of the write-up 10
- In the last review of the semester IV, the marks will be awarded as follows
    - Marks awarded by the supervisor (Considering technical paper writing) 30
    - Marks awarded by the review committee 20

NOTE: A candidate needs to secure a minimum of 50 % marks to be declared to have completed the audit course.

Review/Progress monitoring committee may consider the following points during the assessment.

- In the semester III, the entire design proposal shall be ready, including components/system selection as well as the cost analysis. Two reviews will be conducted based on the presentation given by the student's team.
    - First shall be for finalization of the product selected.
    - Second shall be on finalization of the proposed design of the product.
- In the semester IV, the expected work shall be procurement of components/systems, building of the working prototype, testing and validation of the results based on work completed in semester III.
    - First review is based on readiness of building the working prototype.
    - Second review shall be based on a presentation as well as the demonstration of the working model,during the last month of semester IV. This review will also look at the readiness of the proposed technical paper presentation of the team.

The overall work done by the team shall be assessed based on the following criteria:

1. Quality of survey/ need identification of the product.
2. Clarity of Problem definition (design and development) based on need.
3. Innovativeness in the proposed design.
4. Feasibility of the proposed design and selection of the best solution.
5. Cost effectiveness of the product.
6. Societal impact of the product.
7. Functioning of the working model as per stated requirements.
8. Effective use of standard engineering norms.
9. Contribution of each individual as a member or the team leader.
10. Clarity on the write-up and the technical paper prepared.

The semester reviews (III and IV) may be based on relevant points listed above, as applicable.

Guidelines for Assessment of Semester Reviews:

- The write-up should be prepared as per the guidelines given by the department.
- The design and the development of the product shall be assessed through a presentation and demonstration of the working model by the student team to a panel of Internal and External Examiners, preferably from industryor any research organizations having an experience of more than five years, approved by the Head of the Institution. The presence of the external examiner is desirable only for the 2nd presentation in semester IV. Students are compulsorily required to present the outline of the technical paper prepared by them during the final review in semester IV.

Program: Information Technology **S.Y B.Tech** **Semester: III**
Course: Constitution of India (DJS22A3)

Course Objectives:

1. To provide basic information about Indian constitution.
2. To identify individual role and ethical responsibility towards society.
3. To understand human rights and its implications.

Course Outcomes: On successful completion of this course, the student should be able to:

1. Have general knowledge and legal literacy and thereby to take up competitive examinations.
2. Understand state and central policies, fundamental duties.
3. Understand Electoral Process, special provisions.
4. Understand powers and functions of Municipalities, Panchayats and Co- Operative Societies.
5. Understand Engineering ethics and responsibilities of Engineers.
6. Understand Engineering Integrity & Reliability.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to the Constitution of India:** The Making of the Constitution and Salient features of the Constitution. Preamble to the Indian Constitution Fundamental Rights & its limitations.	02
2	**Directive Principles of State Policy:** Relevance of Directive Principles State Policy Fundamental Duties. Union Executives – President, Prime Minister Parliament Supreme Court of India.	02
3	**State Executives:** Governor, Chief Minister, State Legislature High Court of State. Electoral Process in India, Amendment Procedures, 42nd, 44th, 74th, 76th, 86th & 91st Amendments.	03
4	**Special Provision**s: Provisions for Backward class section of society, Provision for Women, Children & Backward Classes Emergency Provisions.	02
5	**Human Rights:** Meaning and Definitions, Legislation Specific Themes in Human Rights- Working of National Human Rights Commission in India, Powers and functions of Municipalities, Panchayats and Co–Operative Societies.	02
6	**Scope & Aims of Engineering Ethics:** Responsibility of Engineers Impediments to Responsibility. Risks, Safety and liability of Engineers, Honesty, Integrity & Reliability in Engineering	02

Books Recommended:

Text books:

1. Durga Das Basu, "Introduction to the Constitution on India", Prentice –Hall EEE, 19th / 20th Edition, 2001.
2. Charles E. Haries, Michael S Pritchard and Michael J. Rabins, "Engineering Ethics",6th Edition, 2003.

Reference Books:

1. M. V. Pylee, "An Introduction to Constitution of India", Vikas Publishing, 2002.
2. M. Govindarajan, S. Natarajan and V. S. Senthilkumar, "Engineering Ethics", Prentice – Hall of India Pvt. Ltd., 2004.
3. Brij Kishore Sharma, "Introduction to the Constitution of India", PHI Learning Pvt. Ltd., New Delhi, 2011.
4. Latest Publications of Indian Institute of Human Rights, New Delhi.

Program: Information Technology **S.Y B. Tech Semester: IV**
Course: Applied Mathematics (DJS22ITC401)
Course: Applied Mathematics Tutorial (DJS22ITT401)

***Batch wise tutorials are to be conducted.**

Pre-requisite: Engineering Mathematics – I and Engineering Mathematics – II

Course Objectives: The objective of this course is to inculcate an ability to relate engineering problems to mathematical contexts. To introduce students to the concepts of Number Theory by using different theorems. To cover the basic principles of matrices, probability and random variables. The course also familiarizes students with different methods of solving Linear Programming problems.

Course Outcomes: On successful completion of this course, student should be able to:

1. Apply the concept of Cayley-Hamilton Theorem on problems of linear equations and to understand the concept of eigen values and vectors.
2. Interpret the concepts of divisibility, prime number, congruence and number theorems and practice on linear congruence and quadratic linear congruence.
3. Explain the concept of a random variable and the probability distributions.
4. Use the simplex method and its variations to solve linear programming models, given a basic feasible point.
5.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Matrices:** Eigenvalues, Eigenvectors, Algebraic and Geometric multiplicity of an eigenvalue, Similar matrices, diagonalizable matrix, Cayley-Hamilton theorem (without proof), Functions of square matrix, Singular Value Decomposition (SVD).	**08**
2	**Divisibility and primes::** Divisibility, Euclid's algorithm, greatest common divisors, linear Diophantine equation, primes, testing of primes, Sieve of Eratosthenes, prime number theorem,	**04**
3	**Congruences:** Congruences, Fermat's little theorem, Euler's theorem, linear congruence, computing inverse in congruence, Chinese remainder theorem, quadratic congruences, Legendre symbol.	**04**
4	**Probability:** Bayes' theorem, random variables, discrete and continuous, expectation and variance of random variable, probability mass function, probability density function and cumulative distribution function, moments, moment generating function, standard probability distribution: Binomial, Poisson and Normal (for detail study).	**08**
5	**Operations on One and Multiple Random Variable:** Functions of a random variable and their distribution and density functions. Pairs of random variables, Joint CDF, Joint PDF, Independence, Conditional CDF and PDF, Conditional Expectation. One function of two random variables, two functions of two random variables; joint moments, joint characteristic function, covariance and correlation-independent, uncorrelated and orthogonal random variables.	**07**
6	**Mathematical programming: -** Types of solution, standard and canonical form of Linear programming problem (LPP), basic and feasible solutions, simplex method, artificial variables, Big M-method (method of penalty), duality, dual simplex method	**05**

List of Tutorials:

1. Eigenvalues and eigenvectors
2. Function of square matrix, SVD
3. GCD and Diophantine
4. Congruences
5. Bayes' theorem and Random variables, MGF and Moments
6. Standard distributions
7. Joint PMF
8. Joint PDF
9. Simplex and Big-M
10. Duality and dual Simplex

Tutorials:

Minimum eight tutorials based on syllabus will be conducted. Mini project relevant to the subject may be included, which would help the learner to apply the concept learnt.

Books Recommended:

Textbooks:

1. Seymour Lipschutz and Marc Lipson, "Linear Algebra", 4th Edition, Schaum's outlines, 2008.
2. S. C. Gupta and V. K. Kapoor, "Fundamentals of Mathematical Statistics", 10th Edition, Sultan Chand and Sons, 2020.
3. David M. Burton, "Elementary Number Theory", 7th Edition, Mc Graw Hill, 2017.

Reference Books:

1. Gilbert Strang, "Linear Algebra and its Applications", 4th Edition, Cengage, 2005.
2. Ward Cheney and David Kincaid, "Linear Algebra Theory and Applications", 2nd Edition, Jones & Bartlett Learning, 2011.
3. Seymour Lipschutz and John Schiller, "Introduction to Probability and Statistics", Indian Edition, Schaum's outlines, 2017.
4. Ivan Niven, Herbert S. Zuckerman and Hugh L. Montgomery, "An Introduction to the Theory of Numbers", 5th Edition, Wiley, 2008.

Program: Information Technology **S.Y B. Tech Semester: IV**
Course: Formal Languages and Automata Theory (DJS22ITC402)
Course: Formal Languages and Automata Theory Tutorial (DJS22ITL402)

***Batch wise tutorials are to be conducted.**

Pre-requisite: Knowledge of -
1. Basic Mathematical Fundamentals: Sets, Logic, Relations, Functions.

Course Objectives: The objective of the course is to introduce students to the mathematical foundations of computability theory including automata theory & it's applications; the theory of formal languages and grammars; the notions of decidability and computability. The course also enables students to develop the ability to design formal grammar & abstract computing models for formal languages and appreciate the power and limitations of these models.

Course Outcomes: On successful completion of this course, student should be able to:
1. Design formal grammar
2. Design computational model
3. Apply rigorously formal mathematical methods to prove properties of formal languages
4. Prove that the certain languages are undecidable

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Formal Languages:** Introduction, Chomsky Hierarchy. **Regular Language:** Basic Definition, alphabets and strings. **Regular Expression (RE):** Definition, RE operators, operation on regular language such as concatenation, closure, union, interaction, etc. Construction of RE for Regular Language, Pumping lemma for regular language, closure properties of regular language. **Regular Grammar:** Definition, notations, grammar constituents, Left and Right Linear grammar, construction of LL & RL grammar, equivalence of regular grammar and finite automata.	**06**
2	**Finite Automata (FA):** Basic definition, representation, FA as a language acceptor and verifier, different models such as Deterministic FA (DFA) and Non-deterministic FA (NFA). **DFA:** Formal definition, construction of DFA. **NFA:** Formal definition, construction of NFA. Equivalence of DFA's and NFA's **NFA with $\in$-moves:** Formal definition, $\in$-CLOSURE of a state, construction of NFA with $\in$-moves. Equivalence of NFA's with and without $\in$-moves, Equivalence of NFA's with $\in$-moves and DFA, Construction of NFA with $\in$-moves for RE and Construction of RE from FA. Minimal State Finite Automata: necessity and advantages of minimization, minimization algorithm. **Finite Automata with output:** Basic concept, advantages, different models such as Moore and Mealy machines. Moore m/c: formal definition, construction of different Moore m/c models. Mealy m/c: formal definition, construction of Mealy m/c models (examples). Equivalence of Moore and Mealy m/c. Applications of finite automata: lexical analyzer, text editor.	**10**
3	**Context Free Language (CFL) & Context Free Grammar (CFG):** Definition, notations, construction of CFG for CFL. **Derivation:** left most derivation, right most derivation, derivation tree, ambiguous context free grammar, and removal of ambiguity from CFG.	**08**

	Simplification of CFG: live variable, reachable variable, useful variable, useful and useless production, removal of useless variables and useless productions, Nullable variable, $\in$-production, removal of $\in$-productions, unit production, removal of unit productions. **Normal Forms:** Chomsky normal form, Greibach normal forms.	
4	**Push Down Automata (PDA):** Formal definition, instantaneous description, accepted languages, PDA acceptance by Final State and by Empty Stack, deterministic and non-deterministic PDA, construction of PDA for CFG and CFL, construction of CFG for PDA.	**06**
5	**Turing Machine (TM):** Formal definition, instantaneous description, construction of TM. Variations of Turing machine: Two way infinite tapes, Multi-tape, Multiple tracks, Non-deterministic, multidimensional, Multi-head, Church-Turing thesis.	**08**
6	**Undecidability:** Decidable and undecidable problem. Recursive and recursively enumerable language: definition, properties. Universal Turing Machine (UTM) and an undecidable problem. A non-recursive enumerable language, halting problems, other unsolvable problems about TM. Post's correspondence problems: An instance of PCP, modified version of PCP, Undecidability of PCP, applications of PCP.	**04**

List of Tutorials:

1. Designing RE, RG, RLG and LLG for given Regular Language.
2. Converting RE to NFA, NFA to DFA to Reduced DFA, FA to RE.
3. Designing Moore and Mealy machines.
4. Designing CFG and getting Leftmost and Rightmost derivations from it.
5. Simplification of CFG.
6. Converting CFG to CNF & GNF.
7. Designing Push Down Automata for CFL and CFG.
8. Getting CFG from PDA.
9. Designing Turing Machine.
10. Demonstration of JFLAP tool.

Books Recommended:

Text books:

1. John C. Martin, "Introduction to Languages and Theory of Computation", 4th Edition, Tata McGraw Hill,2011.
2. Kavi Mahesh, "Theory of Computation A Problem Solving Approach", 1st Edition, Wiley India, 2011.

Reference Books:

1. John E. Hopcroft, Jeffrey D. Ullman, Motwani, "Introduction to Automata Theory, Languages and Computation", 3rd Edition, Pearson, 2007.
2. Peter Linz, "An Introduction to Formal Languages and Automata", 3rd Edition, Jones and Bartlett Learning, 2001.
3. Harry R. Lewis, Christos H. Papadimitriou, "Elements of the Theory of Computation", 2nd Edition, PHI, 1998.
4. Michael Sipser, "Introduction to the Theory of Computation", 2nd Edition, Thomson Learning, 2006.
5. Bernard M. Moret, "The Theory of Computation", 1st Edition, Pearson Education, 2002.
6. Daniel I. A. Cohen, "Introduction to Computer Theory", 2nd Edition, Wiley, 2014.
7. J. Richard Buchi, "Finite Automata, Their Algebras and Grammars: Towards a Theory of Formal Expressions", 1st Edition, Springer-Verlag, 1989.
8. McNaughton R, "Elementary Computability, Formal Languages and Automata", Prentice-Hall, 1982.
9. K. L. P. Mishra, N. Chandrasekaran, "Theory of Computer Science", 3rd Edition, PHI, 2008.

Program: Information Technology **S.Y B. Tech Semester: IV**
Course: Design and Analysis of Algorithms (DJS22ITC403)
Course: Design and Analysis of Algorithms Laboratory (DJS22ITL403)

Pre-requisite: Computer Programming, Data structures

Course Objectives: The objective of the course is to introduce important algorithmic design paradigms and approaches for effective problem solving. To analyze the algorithm for its efficiency to show its effectiveness over the others. In addition, the concepts of tractable and intractable problems and the classes P, NP and NP-complete problems will be introduced.

Course Outcomes: On successful completion of this course, student should be able to:

1. Analyze the performance of algorithms using asymptotic analysis.
2. Solve the problem using appropriate algorithmic design techniques.
3. Able to prove that certain problems are NP-Complete.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction:** Introduction to Asymptotic Analysis, Analysis of control statements and loops, solving recurrence relations using tree, substitution, master method, analysis of quick sort and merge sort Problem Solving using divide and conquer algorithm - Max-Min problem, Strassen's Matrix Multiplication.	**08**
2	**Greedy Method:** Introduction, control abstraction, Problem solving using - fractional knapsack problem, activity selection problem, job sequencing with deadline, find and union, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Single source shortest path (Dijkstra's algorithm), coin change problem.	**07**
3	**Dynamic Programming:** Introduction, principle of optimality, Components of dynamic programming, characteristics of dynamic programming, Fibonacci problem, Coin Changing problem, 0/1 knapsack (table and set method), Multistage graphs, All pairs shortest paths (Floyd Warshall Algorithm), Single source shortest path (Bellman-Ford Algorithm), Matrix Chain Multiplication, Optimal binary search tree (OBST-successful and unsuccessful search), Travelling salesperson problem, Johnson' algorithm for Flow shop scheduling, Longest Common Subsequence (LCS).	**10**
4	**Backtracking:** Introduction, Basics of backtracking, N–queen problem, Sum of subsets, Graph coloring, Hamiltonian cycles Generating permutation. **Branch-and-Bound:** Introduction, Control abstraction-LC BB, FIFO BB, LIFO BB, Properties, FIFO BB, LIFO BB, LC BB, Fifteen Puzzle problem, 0/1 Knapsack problem, Travelling Salesman problem, Job Sequencing with Deadline	**07**
5	**String Matching Algorithms:** Introduction, The naive string-matching algorithm, The Rabin Karp algorithm, String matching with finite automata, The Knuth Morris Pratt algorithm	**03**
6	**Basics of Computational Complexity:** Complexity classes: The class P and NP, Polynomial reduction, NP Completeness Problem, NP-Hard Problems, NP Completeness problem using Travelling Salesman problem (TSP), Approximation algorithm using TSP	**04**

List of Laboratory Experiments:

1. Implementation of Min Max algorithm
2. Implementation of Strassen's Matrix Multiplication.
3. Implementation of Karatsuba algorithm for long integer multiplication.
4. Fractional Knapsack implementation using greedy approach.

5. Implementation of Activity selection using greedy approach.
6. Implementation of Kruskal's/ Prim's algorithm using greedy approach.
7. Implementation of job sequencing with deadline using greedy approach.
8. Implementation of other greedy algorithms eg: tree vertex split, subset cover, container loading, coin changing, optimal; merge patterns (Huffman tree).
9. Implementation of Single source shortest path (Dijkstra's algorithm).
10. Implementation of Bellman Ford algorithm using Dynamic programming.
11. Implementation of Longest Common Subsequence algorithm using Dynamic programming.
12. Implementation of Travelling Salesperson problem using Dynamic programming.
13. Implementation of multistage graphs/ all pair shortest path using dynamic programming.
14. Implementation of N-queen problem using Backtracking.
15. Implementation of Knuth Morris Pratt string matching algorithm.

Books Recommended:

Textbooks:

1. S. Sridhar, Design and Analysis of Algorithms, 1st Edition, Oxford Education, 2018.
2. Ellis Horowitz , Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms", 2nd Edition, University Press, 2008.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, "Introduction to Algorithms", 3rd Edition, The MIT Press, 2009.
2. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, "Algorithms", 1st Edition, Tata McGraw- Hill, 2017.
3. S.K. Basu, "Design Methods and Analysis of Algorithm", 2nd Edition, PHI, 2013.
4. John Kleinberg, Eva Tardos, "Algorithm Design", 1st Edition, Pearson, 2008.

Program: Information Technology **S.Y B. Tech Semester: IV**
Course: Computer Networks (DJS22ITC404)
Course: Computer Networks Laboratory (DJS22ITL404)

Pre-requisite: Computer System Basics

Course Objectives: The objective of the course is to introduce the students to the layered approach in communication network. This course aims to provide the students with an introduction to standard client-server based applications. The course will also enable the students to explore the services and protocols of each layer and choose appropriate protocols while sending data from sender to receiver using guided or unguided transmission media.

Course Outcomes: On successful completion of this course, student should be able to:

1. Analyze the different types of topologies, network devices and their functions within a network.
2. Analyze key networking protocols and their hierarchical relationship in the conceptual model like TCP/IP and OSI.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction:** Network devices, Network Types: LAN, MAN, WAN, Network topology, OSI Reference model, TCP/IP suite, Comparison of OSI and TCP/IP.	03
2	**Application Layer:** Introduction: Providing Services, Application layer Paradigms, Client-Server Paradigm: Application Programming Interface, Using Services of the Transport Layer, Standard Client Server applications: World Wide Web and HTTP, FTP, Electronic Mail, TELNET, Secure Shell (SSH), Domain Name System (DNS).	06
3	**Transport Layer:** Transport Layer Protocols: Simple Protocol, Stop-and-Wait Protocol, Go-Back-N Protocol (GBN), Selective-Repeat Protocol, Bidirectional Protocols: Piggybacking, User Datagram Protocol: User Datagram, UDP Services, UDP Applications, Transmission Control Protocol: TCP Services, TCP Features, TCP Segment, A TCP Connection, Flow Control, Error Control, TCP Congestion Control, TCP Timers.	10
4	**Network Layer:** Introduction: Network Layer design issues, Communication Primitives: Unicast, Multicast, Broadcast. IPv4 Addressing (Classful and Classless), Subnetting, Supernetting design problems ,IPv4 Protocol, Network Address Translation (NAT) Routing algorithms : Shortest Path (Dijkastra's), Link state routing, Distance Vector Routing Protocols - ARP,RARP, ICMP, IGMP Congestion control algorithms: Open loop congestion control, Closed loop congestion control, QoS parameters, IPv6 packet format, Transition from IPv4 to IPv6.	09
5	**Data Link Layer:** Two Types of Links, Data Link Control: Framing, Flow and Error Control, Error Detection and Correction ((Hamming Code, CRC, Checksum), Two DLC Protocols: HDLC, PPP, **Medium Access Control Sublayer:** Channel Allocation problem, Multiple Access Protocol (Aloha, Carrier Sense Multiple Access (CSMA/CA, CSMA/CD), Wired LANS: Ethernet, Ethernet Standards, Virtual LANs	09
6	**Physical Layer:** Transmission Media: Guided Media- Twisted pair, Coaxial, Fiber optics, Unguided Media (Wireless): Radio Waves, Microwave, Bluetooth and Infrared.	02

List of Laboratory Experiments:

1. To study basic networking commands like ping, tracert, nslookup, netstat, ARP, RARP, ipconfig, ifconfig, dig, traceroute, nslookup, netstat.
2. Implementation of Specific Network topology with respect to Number of nodes and physical layer configuration

3. To implement Graphical simulation of network with Routing Protocols and traffic consideration (TCP, UDP) using NAM.
4. To install Wireshark and study the packet headers.
5. Implementation of connection oriented client server programming using TCP.
6. Implementation of connectionless client server programming using UDP.
7. Implementation of socket programming to demonstrate fault tolerance.
8. Implementation of Routing Protocol.
9. Implementation of Stop and Wait protocol.
10. CRC / Hamming code implementation.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

Books Recommended:

Text books:

1. Behrouz A. Forouzan, Forouzan Mosharrat, "Computer Networks A Top down Approach", Special Indian Edition, McGraw Hill education, 2017.
2. Andrew S Tanenbaum, "Computer Networks", 5th Edition, Pearson Education, 2013.

Reference Books:

1. Behrouz A. Forouzan, "Data Communications and Networking", 5th Edition, McGraw Hill, 2013.
2. James F. Kurose, K. W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", 7th Edition, Pearson Education, 2017.
3. L. L. Peterson and B. S. Davie, "Computer Networks: A Systems Approach", 4th Edition, Elsevier India, 2007.
4. Mayank Dave, "Computer Networks", Cengage Learning, 2012.
5. Achyut Godbole, Atul Kahate, "Data Communications and Networks", 2nd Edition, McGraw Hill, 2017.

Program: Information Technology **S.Y B. Tech Semester: IV**

Course: Python Programming Laboratory (DJS22ITL405)

Pre-requisite: Structured Programming Approach**,** Java Programming

Course Objectives: The objective of the course is to expose students to a new programming language "Python3", thereby making them familiarized with the sequence data types and their interoperability, various control structures and object-oriented programming in Python. To enable students, develop GUI Applications and Web Applications with database connectivity. To lay the foundation of Machine Learning and Data Science Techniques with visualizations.

Course Outcomes: On successful completion of this course, student should be able to:

1. Develop application with a clean coding standard.
2. Implement basic Machine Learning and Data Science Techniques with visualizations.
3. Work effectively as a member of a team.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Basics of Python**: Numbers in Python, Basic & Built-in Math functions, Number Formats, Strings, Quotes, print() function, range () function, Assigning Values to Names & Changing Data Through Names, Copying Data, Tuples, Lists, Dictionaries, Sets, NumPy Arrays, Strings	**04**
2	**Control Statements and Functions:** If statement, if-elif-else, Repetition using while loop, for loop, Defining a Function, Checking & Setting Your Parameters, Default arguments, Variable length arguments, Defining and calling functions within a function, Layers of Functions, Lambda and Filter, Zip(), Map(), Reduce() function, recursion, Function Decorators	**04**
3	**Object Oriented Programming:** Creating a Class, Self-Variables, Constructors, Types of Methods, Constructors in Inheritance, Polymorphism, The super () Method, Method Resolution Order (MRO), Operator Overloading, Method Overloading & Overriding, Interfaces in Python. **Exceptions Handling**: Exceptions, Exception Handling, Types of Exceptions, The Except Block, The assert Statement, User Defined Exceptions.	**06**
4	**Introduction To Data Science Packages:** Creating Modules and Packages, Documenting & Viewing Module, Basics of Testing Your Modules and Packages, Importing & exporting Modules, Random, Matplotlib, Pandas, SciPy, scikit learn Modules.	**06**
5	**Files Handling:** Types of Files in Python, opening a File, closing a File, Writing Text Files, File content manipulation, working with Binary Files, Appending Text to a File, Reading Text Files, File Exceptions, The with Statement Pickle in Python, ZipFile Module.	**02**
6	**GUI Programming with Database Connectivity:** GUI Programming Toolkits, Creating GUI Widgets with Tkinter, Creating Layouts, Form Components, Dialog Boxes. Types of Databases Used with Python, Database Connectivity with Python, Performing DML operations on database. **Web Development Framework:** Flask -Templates, Flask Template Engine: Jinga, Flask-Jinga Template creation, Rendering a web-based application.	**04**

List of Laboratory Experiments:

1. Write python programs to understand Expressions, Variables, Quotes, Basic Math operations, Basic String Operations & String Methods.
2. Write a Python program to implement functions of List, Tuples, Dictionaries, Arrays / Numpy Array (1D, 2D) applications.
3. Write python programs to demonstrate applications of different decision-making statements.
4. Write a Python program to implement Functions and Recursion, Function decorators
5. Write a Python program to implement Programs based on Lambda, Map, Reduce Functions.
6. Write python programs to implement Classes & objects, Constructors, Inheritance & Polymorphism.
7. Write a Python program to implement data analysis using pandas.
8. Write a Python program to implement basic scientific operations using scipy.
9. Write a Python program to implement data visualizations using matplotlib.
10. Write python programs to implement Exception handling.
11. Write python programs to understand different File handling operations with exception handling.
12. Write python programs to understand GUI designing and database operations (Programs based on GUI designing using Tkinter, Mysql database creation & Database connectivity with DML).
13. Write a Python program to implement Web based application with Flask Framework.
14. Mini Project.

Books Recommended:

Text books:

1. Dr. R. Nageswara Rao, "Core Python Programming", Dreamtech Press, 2nd Edition, Wiley Publication, 2007.
2. James Payne, "Beginning Python: Using Python 2.6 and Python 3.1", 2nd Edition, Wrox Publication, 2010.
3. Magnus Lie Hetland, "Beginning Python from Novice to Professional", 3rd Edition, Apress Publication, 2017.

Reference Books:

1. Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson Publication, 2012.
2. E. Balguruswamy, "Introduction to Computing and Problem-Solving using Python", 1st Edition, McGraw Hill Publication, 2016.
3. "Learn to Master Python", from Star EDU solutions, 2nd Edition, ScriptDemics, 2012

Program: Information Technology **S.Y B. Tech Semester: IV**
Course: Universal Human Values (DJS22IHC1)
Course: Universal Human Values Tutorial (DJS22IHT1)

Course Objectives:

1. Development of a holistic perspective based on self-exploration about themselves (human being), family, society, and nature/existence.
2. Understanding (or developing clarity) of the harmony in the human being, family, society, and nature/existence
3. Strengthening of self-reflection.
4. Development of commitment and courage to act.

Course Outcomes: On successful completion of this course, student should be able to:

1. Become more aware of themselves, and their surroundings (family, society, nature); they would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships andhuman nature in mind. They would have better critical ability.
2. Become sensitive to their commitment towards what they have understood (human values, human relationship,and human society).
3. Apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction:** Need, Basic Guidelines, Content and Process for Value Education Purpose and motivation for the course. Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and Experiential Validation- as the process for self-exploration. Continuous Happiness and Prosperity- A look at basic Human Aspirations. Right understanding, Relationship and Physical Facility- the basic requirements for fulfilment of aspirations of every human being with their correct priority. Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario. Method to fulfil the above human aspirations: understanding and living in harmony at various levels.	05
2	**Understanding Harmony in the Human Being:** Harmony in Myself! Understanding human being as a co-existence of the sentient 'I' and the material 'Body'. Understanding the needs of Self ('I') and 'Body' - happiness and physical facility. Understanding the Body as an instrument of 'I' (I am being the doer, seer and enjoyer). Understanding the characteristics and activities of 'I' and harmony in 'I'. Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail. Programs to ensure Sanyam and Health.	05
3	**Understanding Harmony in the Family:** Harmony in Human-Human Relationship. Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship. Understanding the meaning of Trust; Difference between intention and competence. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship	03
4	**Understanding the harmony in society (society being an extension of family):** Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals. Visualizing a universal harmonious order in society- Undivided Society, Universal Order- from family to world family.	03

5	**Understanding Harmony in Nature and Existence:** Whole existence as Coexistence Understanding the harmony in the Nature. Interconnectedness and mutual fulfilment among the four orders of nature recyclability and self-regulation in nature. Understanding Existence as Co-existence of mutually interacting units in all pervasive space. Holistic perception of harmony at all levels of existence.	**05**
6	**Implications of the above Holistic Understanding of Harmony on Professional Ethics:** Natural acceptance of human values. Definitiveness of Ethical Human Conduct. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order. Competence in professional ethics: a. Ability to utilize the professional competence for augmenting universal human order, b. Ability to identify the scope and characteristics of people friendly and eco-friendly production systems, c. Ability to identify and develop appropriate technologies and management patterns for above production systems. Case studies of typical holistic technologies, management models and production systems. Strategy for transition from the present state to Universal Human Order: a. At the level of individual: as socially and ecologically responsible engineers, technologists, and managers, b. At the level of society: as mutually enriching institutions and organizations.	**05**

Tutorials:

Term work shall consist of minimum 5 activities based on activities conducted.

The tutorials could be conducted as per the following topics: -

Activity No 1	Practice sessions to discuss natural acceptance in human being as the innate acceptance for living with responsibility (living in relationship, harmony, and co-existence) rather than as arbitrariness in choice based on liking-disliking.
Activity No 2	Practice sessions to discuss the role others have played in making material goods available to me. Identifying from one's own life. Differentiate between prosperity and accumulation. Discuss program for ensuring health vs dealing with disease.
Activity No 3	Practice sessions to reflect on relationships in family, hostel and institute as extended family, real life examples, teacher-student relationship, goal of education etc. Gratitude as a universal value in relationships. Discuss with scenarios. Elicit examples from students' lives.
Activity No 4	Practice sessions to discuss human being as cause of imbalance in nature (film "Home" can be used), pollution, depletion of resources and role of technology etc.
Activity No 5	Practice Exercises and Case Studies will be taken up in Practice (tutorial) Sessions e.g. To discuss the conduct as an engineer or scientist etc.

Books Recommended:

Text books:

1. R R Gaur, R Sangal, G P Bagaria, "Human Values and Professional Ethics", Excel Books, New Delhi, 2010.

Reference Books:

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi.
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews.
7. Economy of Permanence - J C Kumarappa.
8. Bharat Mein Angreji Raj – PanditSunderlal.
9. Rediscovering India - by Dharampal.

10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi.
11. India Wins Freedom - Maulana Abdul Kalam Azad.
12. Vivekananda - Romain Rolland. (English)
13. Gandhi - Romain Rolland. (English)

Prepared by Checked by Head of the Department Vice Principal Principal

Program: Information Technology **S.Y B. Tech Semester: IV**
Course: Environmental Studies (DJS22A4)

Pre-requisite: Interest in Environment and its impact on Human

Course Objectives:

1. Understand environmental issues such as depleting resources, pollution, ecological problems and the renewable energy scenario.
2. Familiarize environment related legislation

Course Outcomes: On successful completion of this course, student should be able to:

1. Understand how human activities affect environment.
2. Understand the various technology options that can make a difference.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Social Issues and Environment:** Ecological footprint and Carrying Capacity, Depleting nature of Environmental resources such as soil, water minerals and forests, Carbon emissions and Global Warming.	**04**
2	**Technological Growth for Sustainable Development:** Social, Economic and Environmental aspects of Sustainable Development, Renewable Energy Harvesting, Concept of Carbon credit, Green Building, Power and functions of Central Pollution Control Board and State Pollution Control Board .	**04**
3	**Green Technology:** History, Agenda, and Challenges Ahead. Sustainable Cloud Computing, and Risk Management, Sustainable Software Design, Data Center Energy Efficiency, Thin-Client and Energy Efficiency.	**05**

Books Recommended:

Text books:

1. R. Rajagopalan, "Environmental Studies From Crisis to Cure", 3rd Edition, Oxford University Press India, 2011.
2. Erach Bharucha, "Textbook for Environmental Studies For Undergraduate Courses of all Branches of Higher Education".
3. Mohammad Dastbaz, Colin Pattinson, Babak Akhgar, Morgan and Kaufman, "Green Information TechnologyA Sustainable Approach", 1st Edition, Elsevier, 2015.

Reference Books:

1. Paulina Golinska, Marek Fortsch, Jorge Marx-Gómez, "Information Technologies in Environmental Engineering: New Trends and Challenges", Springer, 2011.

Program: Information Technology **S.Y B. Tech** **Semester: IV**
Course: IPD II (DJS22A5)

Course Objectives:

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuring that a solution is found out to address the same by designing and developing an innovative product.
2. To familiarize the students with the process of designing and developing a product, while they work as part ofa team.
3. To acquaint the students with the process of applying basic engineering fundamentals, so as to attempt at the design and development of a successful value added product.
4. To inculcate the basic concepts of entrepreneurship and the process of self-learning and research required to conceptualize and create a successful product.

Course Outcome: On successful completion of this course, student should be able to:

1. Identify the requirement for a product based on societal/research needs.
2. Apply knowledge and skills required to solve a societal need by conceptualizing a product, especially while working in a team.
3. Use standard norms of engineering concepts/practices in the design and development of an innovative product.
4. Draw proper inferences through theoretical/ experimental/simulations and analyze the impact of the proposed method of design and development of the product.
5. Develop interpersonal skills, while working as a member of the team or as the leader.
6. Demonstrate capabilities of self-learning as part of the team, leading to life-long learning, which could eventually prepare them to be successful entrepreneurs.
7. Demonstrate product/project management principles during the design and development work and also excel in written (Technical paper preparation) as well as oral communication.

Guidelines for the proposed product design and development:

- Students shall form a team of 3 to 4 students (max allowed: 5-6 in extraordinary cases, subject to the approvalof the department review committee and the Head of the department).
- Students should carry out a survey and identify the need, which shall be converted into conceptualization of a product, in consultation with the faculty supervisor/head of department/internal committee of faculty members.
- Students in the team shall understand the effective need for product development and accordingly select the best possible design in consultation with the faculty supervisor.
- Students shall convert the best design solution into a working model, using various components drawn from their domain as well as related interdisciplinary areas.
- Faculty supervisor may provide inputs to students during the entire span of the activity, spread over 2 semesters, wherein the main focus shall be on self-learning.
- A record in the form of an activity log-book is to be prepared by each team, wherein the team can record weekly progress of work. The guide/supervisor should verify the recorded notes/comments and approve the same on a weekly basis.
- The design solution is to be validated with proper justification and the report is to be compiled in a standard format and submitted to the department. Efforts are to be made by the students to try and publish a technical paper, either in the institute journal, “Techno Focus: Journal for Budding Engineers” or at a suitable publication,approved by the department research committee/ Head of the department.
- The focus should be on self-learning, capability to design and innovate new products as well as on developing the ability to address societal problems. Advancement of entrepreneurial capabilities and quality development of the students through the year long course should ensure that the design and development of a product of appropriate level and quality is carried out, spread over two semesters, ie during the semesters III and IV.

Guidelines for Assessment of the work:

- The review/ progress monitoring committee shall be constituted by the Head of the Department. The progress of

design and development of the product is to be evaluated on a continuous basis, holding a minimum of two reviews in each semester.

- In the continuous assessment, focus shall also be on each individual student's contribution to the team activity, their understanding and involvement as well as responses to the questions being raised at all points in time.
- Distribution of marks individually for the both reviews as well as for the first review during the subsequent semester shall be as given below:
    - Marks awarded by the supervisor based on log-book 20
    - Marks awarded by review committee 20
    - Quality of the write-up 10

  In the last review of the semester IV, the marks will be awarded as follows
    - Marks awarded by the supervisor (Considering technical paper writing)30
    - Marks awarded by the review committee 20

NOTE: A candidate needs to secure a minimum of 50 % marks to be declared to have completed the audit course.

Review/progress monitoring committee may consider the following points during the assessment.

- In the semester III, the entire design proposal shall be ready, including components/system selection as well as the cost analysis. Two reviews will be conducted based on the presentation given by the student's team.
    - First shall be for finalization of the product selected.
    - Second shall be on finalization of the proposed design of the product.
- In the semester IV, the expected work shall be procurement of components/systems, building of the working prototype, testing and validation of the results based on work completed in semester III.
    - First review is based on readiness of building the working prototype.
    - Second review shall be based on a presentation as well as the demonstration of the working model, during the last month of semester IV. This review will also look at the readiness of the proposed technical paper presentation of the team.

The overall work done by the team shall be assessed based on the following criteria;

1. Quality of survey/ need identification of the product.
2. Clarity of Problem definition (design and development) based on need.
3. Innovativeness in the proposed design.
4. Feasibility of the proposed design and selection of the best solution.
5. Cost effectiveness of the product.
6. Societal impact of the product.
7. Functioning of the working model as per stated requirements.
8. Effective use of standard engineering norms.
9. Contribution of each individual as a member or the team leader.
10. Clarity on the write-up and the technical paper prepared.

The semester reviews (III and IV) may be based on relevant points listed above, as applicable.

Guidelines for Assessment of Semester Reviews:

- The write-up should be prepared as per the guidelines given by the department.
- The design and the development of the product shall be assessed through a presentation and demonstration of the working model by the student team to a panel of Internal and External Examiners, preferably from industryor any research organizations having an experience of more than five years, approved by the Head of the Institution. The presence of the external examiner is desirable only for the 2nd presentation in semester IV. Students are compulsorily required to present the outline of the technical paper prepared by them during the final review in semester IV.

Prepared by Checked by Head of the Department Vice Principal Principal