MA23C09 FINITE STATE AUTOMATA AND DISCRETE STRUCTURES

L	T	P	C
3	1	0	4

OBJECTIVES:

- The students must be able to understand mathematical logic and to develop analytical solutions for logical problems.
- Apply graph model and graph techniques for solving network connectivity and other problems.
- Students will be able to comprehend the algebraic structure and formal languages with their applications to handle abstract generalizations.
- To introduce finite state automata as language acceptor of regular sets.
- To introduce context free grammars and context free languages and their normal forms.

UNIT I LOGIC 9+3

Statements – Connectives – Truth Tables – Normal Forms – Predicate Calculus – Methods of proof – Inference Theory - Mathematical Induction.

UNIT II GRAPHS 9+3

Graphs and Graph Models – Graph Terminology and Special types of Graphs – Matrix Representation of Graphs and Graph Isomorphism – Connectivity – Euler and Hamiltonian Paths.

UNIT III ALGEBRAIC STRUCTURES 9+3

Groups – Cyclic group – Permutation group – Substructures – Homomorphism – Cosets and Lagrange's Theorem – Normal Subgroups – Rings and Fields (definition and examples).

UNIT IV FINITE STATE AUTOMATA 9+3

Finite state automata – Deterministic and non-deterministic model – Languages accepted by Finite State Automata – Regular expressions and Regular sets – Pumping lemma for regular sets.

UNIT V CONTEXT FREE GRAMMER 9+3

Grammar - Context-free Grammars - Derivation trees - Simplification of context free grammar (only Construction and no proof of equivalence of grammars) - Chomsky normal Form - Greibach Normal Form – Pumping lemma for context-free languages.

TOTAL: 60 PERIODS

Laboratory based exercises / assignments / assessments will be given to students from the content of the course wherever applicable.

Branch specific / General Engineering applications based on the content of each units will be introduced to students wherever possible.

Suggested Laboratory based exercises / assignments / assessments : (IST)

Logic

1. Construction of truth table for a given statement formula with three variables, checking satisfiability of the statement formula with three variables.
2. Construct PDNF and PCNF for a given statement formula with three variables.

Graphs

1. Checking graph isomorphism using adjacency matrix.

2. Finding the shortest path in a connected weighted graph (Dijkstra's algorithm).

Algebraic Structures

1. Modular exponentiation.
2. Euclidean algorithm.(Ref. Rosen pg. 226 – 227).

Finite State Automata

1. Construction of finite state automaton for a given regular set.
2. Finding language accepted by a given finite state automaton.

Grammars

1. Finding the language generated by a given context-free grammar.
2. Construction of a context-free grammar for generating a given context-free language.

OUTCOMES:

CO1 : The students are able to apply mathematical logic and to find analytical solutions for logical problems.

CO2 : The students are able to apply graph model and graph techniques for solving network connectivity and other problems.

CO3 : Students will be able to apply the algebraic structure and formal languages with their applications to handle abstract generalizations.

CO4 : Students will be able to design finite state automata to accept regular sets.

CO5 : Students will be able to form context-free grammar to generate context-free language.

TEXT BOOKS:

1. Trembley.J.P. and Manohar R. “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw – Hill Publishing Company Limited, New Delhi. Reprinted in 2007. (For Unit I, III, IV)
2. Hopcroft, J.E., Rajeev Motwani and Ullman, J.D. “Introduction to Automata Theory,Languages, and Computation”, Pearson Education, Second Edition, Harlow, 2014.

REFERENCES:

1. Kenneth H. Rosen, “Discrete Mathematics and its Applications”, Tata McGraw Hill Publishing Company Limited, New Delhi. Reprinted in 2007 (6th Edition).
2. Hopcroft J.E. and Ullman J.D. “Introduction to Automata Theory, Languages and Computation”, Narosa Publishing House,2002.
3. Thomas Koshy, “Discrete Mathematics with Applications”, Academic Press, Reprinted in 2005.

CO – PO Mapping:

Course Outcomes	PROGRAMME OUTCOMES											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**P10**	**P11**	**P12**
CO1 :	3	3	2	3	1	2	1	1	1	1	1	3
CO2 :	3	3	2	3	1	2	1	1	1	1	1	3
CO3 :	3	3	2	3	1	2	1	1	1	1	1	3
CO4 :	3	3	2	3	1	2	1	1	1	1	1	3
CO5 :	3	3	2	3	1	2	1	1	1	1	1	3

IT23301	DIGITAL LOGIC AND DESIGN	L	T	P	C
		3	0	2	4

UNIT I BOOLEAN ALGEBRA AND GATES 9

Number Systems: Binary, Octal, Hexadecimal – Representation of Negative Numbers – Complements – Arithmetic Operations – Binary Codes – Boolean Algebra – Theorems and Postulates – Functions – Truth Table – Logic Gates – Universal gates– Canonical and Standard Forms – Minterms and Maxterms – Sum of Products and Product of Sums.

UNIT II KARNAUGH MAP AND COMBINATIONAL LOGIC 9

Simplification of Boolean Functions –Karnaugh Map – 2, 3, 4 variable- Don't-care conditions, Prime and essential prime Implicants – NAND/NOR Implementations – Combinational Circuits – Arithmetic Circuits – Half and Full Adders – Subtractors – Introduction to HDL.

UNIT III COMBINATIONAL LOGIC 9

Design procedure, Binary Parallel Adder and Subtractors- Carry Look-ahead Adder – BCD Adder – Binary Multiplier – Magnitude Comparator – Code Converters – Decoder – Encoder – Priority Encoder – Multiplexers - Demultiplexers – Applications.

UNIT IV SEQUENTIAL LOGIC 9

Sequential Circuits- Latches, flip-flops- Characteristic tables and excitation tables – Analysis of clocked sequential circuits – Moore /Mealy models – Registers: Shift Registers, Universal Shift Register – Counters – Asynchronous Ripple Counters - Synchronous Counters- ring Counter- Johnson Counter.

UNIT V PROGRAMMBLE LOGIC DEVICES 9

Memory Systems – RAM – ROM – Memory Decoding – Error detection and correction - Checksum - Digital System Design using PROM – PLDs - Programmable Logic Array - Programmable Array Logic - CPLDs - Field Programmable Gate Array.

THEORY: 45 PERIODS

EXERCISES 30

1. Verification of Boolean theorems using logic gates.
2. Design and implementation of combinational circuits using gates for arbitrary functions.
3. Implementation of 4-bit binary adder/ subtractor circuits and getting started with HDL.
4. Implementation of combinational circuits using code converters.
5. Implementation of BCD adder, encoder, and decoder circuits.
6. Implementation of any one of the synchronous counters.
7. Implementation of a Universal Shift register.
8. HDL coding for any of the combinational and sequential circuits.
9. Mini project on the design of a digital circuit for solving practical problems.

TOTAL: 75 PERIODS

COURSE OUTCOMES

Upon successful completion of the course, the student will be able to:

1. Simplify complex Boolean functions.
2. Implement digital circuits using simplified methods and combinational logic ICs.
3. Design digital circuits with various combinational logic and write HDL for digital system.
4. Understand the characteristics of various sequential circuits with combinational circuits.
5. Design and implement various programmable logic devices.

TEXT BOOKS:

1. M. Morris Mano, Michael D. Ciletti, “Digital Design”, Sixth Edition, Pearson Education, 2018.

REFERENCES:

1. Charles H. Roth Jr., “Fundamentals of Logic Design”, Fifth Edition, Jaico Publishing House, 2003.
2. John F. Wakerly, “Digital Design Principles and Practices”, Fourth Edition, Pearson Education, 2007.
3. Donald D. Givone, “Digital Principles and Design”, Tata McGraw Hill, 2003.
4. G. K. Kharate, “Digital Electronics”, Oxford University Press, 2010.
5. Harris, Sarah, and David Harris. Digital Design and Computer Architecture, RISC-V Edition. Morgan Kaufmann, 2021.

COURSE OUTCOMES	Program Outcomes (POs) & Program Specific Outcomes (PSOs)														
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	3	3	3	2	-	-	-	-	1	-	-	2	3	3	3
CO2	3	3	3	2	-	-	-	-	2	-	-	2	3	3	3
CO3	3	3	3	2	1	1	1	-	2	-	-	2	3	3	3
CO4	3	3	3	3	-	-	-	-	2	1	1	2	3	3	3
CO5	3	3	3	3	2	-	-	-	2	1	1	2	3	3	3
AVG	**3**	**3**	**3**	**2.4**	**1.5**	**1**	**1**	**-**	**1.8**	**1**	**1**	**2**	**3**	**3**	**3**

1-low, 2-medium, 3-high, ‘-“- no correlation

IT23302	DATA STRUCTURES	L	T	P	C
		3	0	2	4

UNIT I INTRODUCTION TO DATA STRUCTURES 9

Overview of Arrays, Functions, Structures, Pointers – Classification of Data Structures- Operations on Data Structures - Abstract Data Types (ADTs) – Introduction to Time and Space Complexity- Searching Techniques – Sorting: Selection Sort- Insertion Sort – Radix Sort- Linear Sort: Counting Sort- External Sorting.

UNIT II LINEAR DATA STRUCTURES 9

List ADT – Array-Based Implementation – Linked List – Doubly-Linked Lists – Circular Linked List – Stack ADT – Applications of Stack: Infix to Postfix Conversion- Evaluation of Postfix expression- Recursion: Tower of Hanoi - Queue ADT – Linear Queue – Circular Queue – Dequeue.

UNIT III TREES 9

Introduction to Trees – Binary Trees – Tree Traversals: Inorder – Preorder- Postorder Traversals – Expression Trees – Binary Search Tree ADT- Operations: Insert- Delete - Applications of Trees- Priority Queues: Binary Heap : Properties- Operations: Insert- Findmin and Findmax- DeleteMin- Applications of Binary Heap – Heap Sort.

UNIT IV GRAPHS 9

Introduction to Graphs – Properties – Representation of Graphs – Graphs Traversals: Breadth First Search and Depth First Search – Topological Sort – Shortest path algorithm: Unweighted Shortest path – Dijkstra's algorithm – Minimum Spanning Tree: Prims algorithm – Kruskal's algorithm.

UNIT V HASHING TECHNIQUES 9

Hashing- Hash Table- Hash Functions: Division Method- Multiplication method- Mid square method- Folding method – Collision Resolution by Separate Chaining – Collision Resolution through Open Addressing: Linear Probing– Quadratic Probing – Double Hashing – Rehashing – Extendible Hashing – Applications of Hashing.

THEORY: 45 PERIODS

EXERCISES

- Practice of C Programming in solving real time problems using Structures, arrays, functions, pointers and Preprocessor Directives.
- Implementation of Array ADT using Linear Search and Binary Search.
- Implementation of Insertion Sort, Quick Sort, Merge Sort.
- Implementation of Linked List ADT.
- Implementation of Stack ADT using Arrays and Linked List.
- Implementation of Queue ADT using Arrays and Linked List.
- Implementation of Stack applications.
- Implementation of Binary Search Tree ADT with Tree Traversals.
- Implementation of Priority Queue ADT with Heap Sort.
- Implementation of Graph, Graph Traversals and Topological Sort.
- Implementation of Shortest path using Dijkstras Algorithm.

30

- Implementation of Spanning Tree using Prims Algorithm.
- Implementation of Hashing using Open Addressing technique.

TOTAL : 75 PERIODS

COURSE OUTCOMES:

Upon successful completion of the course, the student will be able to:

CO1: Implement sort and search algorithms appropriately for a given application using Array ADT.
CO2: Analyze and apply suitable linear data structures for efficient data storage.
CO3: Analyze and use appropriate tree data structure operations for storage and faster access.
CO4: Understand the usage of Graph data structures to solve a real time problem.
CO5: Understand and apply appropriate hash functions that result in a collision free scenario for data storage and retrieval.

TEXT BOOKS:

1. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, Second Edition, Pearson Education, 2014.
2. Reema Thareja, “Data Structures using C”, Third Edition, Oxford University Press, 2023.

REFERENCES:

1. Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015.
2. Paul J. Deitel, Harvey Deitel, “C How to Program”, Seventh Edition, Pearson Education, 2013.
3. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education,1983.
4. Ellis Horowitz, Sartaj Sahni and Susan Anderson, “Fundamentals of Data Structures”, Galgotia, 2008.

COURSE OUTCOMES	Program Outcomes (POs) & Program Specific Outcomes (PSOs)														
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	3	3	3	3	3	-	-	-	-	-	1	3	3	3	3
CO2	3	3	3	3	2	-	-	-	-	-	1	3	3	3	3
CO3	3	3	3	3	2	-	-	-	-	-	1	3	3	3	3
CO4	3	3	3	3	2	-	-	-	-	-	1	3	3	3	3
CO5	3	3	3	3	2	-	-	-	-	-	1	3	3	3	3
AVG	**3**	**3**	**3**	**3**	**2.2**	**-**	**-**	**-**	**-**	**-**	**1**	**3**	**3**	**3**	**3**

1-low, 2-medium, 3-high, ‘-“- no correlation

IT23303 **DATABASE MANAGEMENT SYSTEMS**

L	T	P	C
3	0	2	4

UNIT I RELATIONAL DATABASES **9**

Purpose of Database System – Views of Data – Data Models – Database System Architecture – Introduction to Relational Databases – Relational Model – Keys – Relational Algebra – Relational Calculus – SQL Fundamentals – Advanced SQL features – Triggers – Embedded SQL.

UNIT II DATABASE DESIGN **9**

Entity-Relationship Model – ER Diagrams – Functional Dependencies – Non-Loss Decomposition Functional Dependencies – First Normal Form – Second Normal Form – Third Normal Form – Dependency Preservation – Boyce/Codd Normal Form – Multi-Valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form.

UNIT III TRANSACTION MANAGEMENT **9**

Transaction Concepts – ACID Properties – Serializability – Transaction Isolation Levels – Concurrency Control – Need for Concurrency – Lock-Based Protocols - Timestamp-Based Protocols – Deadlock Handling – Recovery System – Failure Classification – Recovery Algorithm - ARIES.

UNIT IV IMPLEMENTATION TECHNIQUES **9**

Overview of Physical Storage Media – RAID – File Organization – Organization of Records in Files – Indexing and Hashing – Ordered Indices – B+ tree Index Files – Static Hashing – Dynamic Hashing – Query Processing Overview – Catalog Information for Cost Estimation – Query Optimization.

UNIT V ADVANCED TOPICS **9**

Overview of Distributed Databases – Data Fragmentation – Replication – NOSQL Database: Characteristics – CAP theorem – Types of NoSQL Datastores: Column Oriented, Document, Key-Value and Graph Types – Introduction to MongoDB – Data Model - JSON and BSON - Polymorphic Schemas - Basic Querying.

THEORY: 45 PERIODS

EXERCISES **30**

1. Create a database table, add constraints (primary key, unique, check, NOT NULL), insert rows, update, and delete rows using SQL DDL and DML commands.
2. Create set of tables, add foreign key constraints, and incorporate referential integrity.
3. Query the database tables using different ‘where’ clause conditions and implement aggregate functions.
4. Query the database tables and explore sub queries and simple join operations.
5. Query the database tables and explore natural, equi, and outer joins.
6. Write user defined functions and stored procedures in SQL.
7. Execute complex transactions and realize DCL and TCL commands.
8. Write SQL Triggers for insert, delete, and update operations in database table.
9. Create View and index for database tables with large number of records.
10. Create Document, column, and document - based data using NOSQL database tools.

11. Develop a simple GUI based database application and incorporate all the above-mentioned features.

TOTAL: 75 PERIODS

COURSE OUTCOMES

Upon successful completion of the course, the student will be able to:

1. Understand the key principles, the structures, and the organization of relational databases and to formulate query using relational algebra/ SQL.
2. Identify the methodology of conceptual modelling through ER Model and use formal techniques like normalization to design a database schema.
3. Demonstrate the transactions and estimate the procedures for controlling the consequences of concurrent data access.
4. Analyze the database storage structures, access and query processing techniques.
5. Understand and differentiate the principles and common features of the distributed, and NoSQL databases.

TEXT BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Seventh Edition, Tata McGraw Hill, 2020.
2. Shakuntala Gupta Edward and Navin Sabharwal, “Practical MongoDB: Architecting, Developing, and Administering MongoDB”, Apress, 2015.

REFERENCES:

1. C. J. Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.
2. Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, Fourth Edition, Tata McGraw Hill, 2010.
3. Carlos Coronel, Steven Morris, Peter Rob, “Database Systems: Design, Implementation and Management”, Twelfth Edition, Cengage Learning, 2017.

COURSE OUTCOMES	Program Outcomes (POs) & Program Specific Outcomes (PSOs)														
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	3	2	2	2	2	-	-	-	2	-	1	2	3	3	3
CO2	2	3	2	2	2	-	-	-	2	-	1	2	3	3	3
CO3	2	3	3	2	3	-	-	-	2	-	1	2	2	2	2
CO4	1	3	2	3	2	-	-	-	3	-	2	2	3	3	3
CO5	1	2	2	2	2	-	-	-	2	-	1	2	2	2	2
AVG	**1.8**	**2.6**	**2.2**	**2.2**	**2.2**	**-**	-	**-**	**2.2**	**-**	**1.2**	2	**2.6**	**2.6**	**2.6**

1-low, 2-medium, 3-high, ‘-“- no correlation

IT23304 **OBJECT ORIENTED PROGRAMMING**

L	T	P	C
1	0	2	2

UNIT I OVERVIEW OF OOP, CLASS AND OBJECTS **3**

Object Oriented Programming Concepts – Procedure vs. Object-oriented programming –Tokens - Pointers - User-defined types – ADT- Classes and Objects- Member Functions – Data Members- private and public members – static, Inline, friend and constant Functions – Constructors and Destructors - this Pointer.

UNIT II OVERLOADING **3**

Function Overloading - Operator Overloading – Fundamentals – Restrictions – Operator functions as Class members vs Global Functions – Overloading stream insertion and Stream extraction operators – Unary – Binary operator overloading - Dynamic Memory Management.

UNIT III INHERITANCE AND POLYMORPHISM **4**

Inheritance -types– Base and derived classes - protected members -Relationship between base class and derived classes with case study - private, public and protected inheritance- Constructors and Destructors in Derived Classes – Polymorphism - Relationships among Objects in an Inheritance Hierarchy – Compile time vs Runtime Polymorphism - Virtual Functions – Abstract Classes – Pure Virtual Functions.

UNIT IV TEMPLATES AND STANDARD TEMPLATE LIBRARY **3**

Function Template – Overloading Function Templates - Class Template – Non Type parameters and Default types for Class Templates – Templates and Inheritance, friend and Static Members - Name spaces- Casting- Standard Template Library – Container Classes – Vectors – Lists – Maps- Strings.

UNIT V I/O SYSTEM, FILE I/O AND EXCEPTION HANDLING **2**

C++ Streams - C++ Stream classes – Formatted IO – File classes and File operations - Case Study - Exception Handling –User defined Exceptions - try, catch, throw - rethrowing an Exception – Standard Library Exception Hierarchy.

THEORY: 15 PERIODS

EXERCISES **30**

1. Programs using Data types, Operators and Control Structures.
2. Programs using Arrays and Strings.
3. Programs using Functions and Pointers.
4. Programs using User-defined types.
5. Programs using Classes and Objects.
6. Programs using Constructors and Destructors.
7. Programs using Operator Overloading.
8. Programs using Inheritance, Polymorphism and its types.
9. Programs using Dynamic memory allocation.
10. Programs using Templates and Exceptions.
11. Programs using Sequential and Random access files.
12. Programs using Standard Template Library .

TOTAL: 45 PERIODS

COURSE OUTCOMES

Upon successful completion of the course, the student will be able to:

CO1: Understand the Object-oriented programming concepts and fundamentals.
CO2: Implement the features of overloading in object oriented programming.
CO3: Implement the concept of reusability and polymorphism.
CO4: Write generic programs and STL based applications.
CO5: Create and process data in files using file I/O functions with exception handling.

TEXT BOOKS

1. HM Deitel and PJ Deitel, “C++ How to Program”, Tenth Edition, Pearson Education, 2020.
2. Herbert Schildt, “The Complete Reference in C++”, Fifth Edition, Tata McGraw Hill, 2017(Reprint).

REFERENCES

1. Bjarne Stroustrup, “The C++ Programming language”, Fourth edition, Pearson Education, 2013.
2. Stephen Prata, “C++ Primer Plus”, Sixth Edition, Pearson Education, 2011.
3. E Balagurusamy, “Object oriented Programming with C++”, Eighth edition, Tata McGraw Hill, 2020.
4. Marc Gregoire, “Professional C++”, 5th Edition, Wrox, 2021.

COURSE OUTCOMES	**Program Outcomes (POs) & Program Specific Outcomes (PSOs)**														
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	2	3	3	3	3	-	-	-	1	-	3	2	3	3	3
CO2	2	3	3	3	3	-	-	-	1	-	3	2	3	3	3
CO3	2	3	3	3	3	-	-	-	1	-	3	2	3	3	3
CO4	2	3	3	3	3	-	-	-	1	-	3	2	3	3	3
CO5	2	3	3	3	3	-	-	-	1	-	3	2	3	3	3
AVG	**2**	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**1**	**-**	**3**	**2**	**3**	**3**	**3**

1-low, 2-medium, 3-high, ‘-“- no correlation

IT23U01	STANDARDS – IT	L	T	P	C
		1	0	0	1

MODULE I – OVERVIEW OF STANDARDS **6**

Basic concepts of standardization: Purpose of Standardization, marking and certification of articles and processes; Importance of standards to industry, policy makers, trade, sustainability and innovation. Objectives, roles and functions of BIS, Bureau of Indian Standards Act, ISO/IEC Directives; WTO Good Practices for Standardization. Important Indian and International Standards.

MODULE II INTERNATIONAL STANDARDS IN COMPUTER SCIENCE **9**

Introduction -Importance of standards in IT-Overview of key international standards organizations **ANSI and IEEE Standards** - ANSI standards for software engineering (e.g., ANSI/ISO/IEC 12207:2008 - Software Life Cycle Processes)- IEEE standards and their applications in software engineering (e.g., IEEE 830-1998 - Requirements Specifications)- **ISO/IEC 20000**: IT Service Management -Scope and requirements-Service delivery process-Certification and implementation challenges- ISO 9000 Series: Quality Management - Overview of ISO 9001-Quality management principles-Certification process and benefits-

ITU-T Standards in Telecommunications-Overview of ITU-T series (e.g., ITU-T X.509 for public key infrastructure)-Impact on global telecommunications standards- **IETF Standards in Internet Protocols**-Overview of key IETF standards (e.g., RFC 791 for IPv4)-Evolution and adoption of internet protocols-**W3C Standards for the World Wide Web** -Key W3C standards (e.g., HTML5, CSS3, Web Accessibility Guidelines)-Role of standards in web development and interoperability

ISO/IEC 27001: Information Security Management -Principles and Framework-Risk assessment and Management-Controls and compliance-**NIST Standards and Frameworks** - NIST Cybersecurity Framework (CSF)NIST Special Publications (e.g., SP 800 series) for cybersecurity **ACM Standards and Guidelines** -ACM Code of Ethics and Professional Conduct-ACM Computing Classification System (CCS) and its role in standardization

Total : 15 PERIODS

REFERENCES:

1. Manual for Standards Formulation 2022, Bureau of Indian Standards
2. Kunas, Michael, “Implementing service quality based on ISO/IEC 20000: A management guide” IT Governance publishing, 2012.
3. Kan, S. H. “Standards for Information Technology and Systems”, Prentice Hall, 2017.
4. IEEE Computer Society. (2014) “IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK)”, Version 3.0. IEEE. Retrieved from IEEE Xplore
5. Calder, Alan. “ISO/IEC 27001:2013 – A Pocket Guide” IT Governance Publishing, 2013.
6. Sikos, Leslie,” Web Standards: Mastering HTML5, CSS3, and XML.” Apress, 2011.
7. Association for Computing Machinery. “ACM Code of Ethics and Professional Conduct: A Guide” ACM, 2018
8. Calder, Alan, “NIST Cybersecurity Framework: A Pocket Guide. IT Governance Publishing” 2018.

UC23U01 **UNIVERSAL HUMAN VALUES** **L T P C**
1 0 2 2

COURSE OBJECTIVE:

The objective of the course is four-fold:

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.

MODULE I: INTRODUCTION (3L,6P)

Purpose and motivation for the course, recapitulation from Universal Human Values-I, Self-Exploration– 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.

Practical Session: *Include 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*

MODULE II: HARMONY IN THE HUMAN BEING (3L,6P)

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 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.

Practical Session: *Include 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.*

MODULE III: HARMONY IN THE FAMILY AND SOCIETY (3L,6P)

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, Understanding the harmony in the 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.

Practical Session*: Include 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*

MODULE IV: HARMONY IN THE NATURE AND EXISTENCE (3L,6P)

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.

Practical Session: *Include 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.*

MODULE V: IMPLICATIONS OF HARMONY ON PROFESSIONAL ETHICS (3L,6P)

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, Sum up.
Practical Session: *Include Exercises and Case Studies will be taken up in Sessions E.g. To discuss the conduct as an engineer or scientist etc.*

TOTAL: 45 (15 Lectures + 30 Practicals) PERIODS

COURSE OUTCOME:

By the end of the course, the students will be able to:

1. Become more aware of themselves, and their surroundings (family, society, nature);
2. Have more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.
3. Have better critical ability.
4. Become sensitive to their commitment towards what they have understood (human values, human relationship and human society).
5. 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.

REFERENCES:

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 3rd revised edition, 2023.
2. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
3. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
4. The Story of Stuff (Book).
5. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi