R.V.R & J.C. College of Engineering (Autonomous)

Department of CSE (Data Science)

B.Tech – IV Semester [Second Year]

1	CD221	Foundations of Artificial Intelligence
2	CD222	Database Management Systems
3	CD223	Operating Systems
4	CD224	Software Engineering
5	CD225	Design and Analysis of Algorithms
6	CD261	Database Management Systems Lab
7	CD262	Operating Systems & Software Engineering Lab
8	CD263	Design and Analysis of Algorithms Lab
9	CDSL2	Scripting Languages (Skill Oriented Course)
10	CDMC4	Ethics & Human Values

CD221	Foundations of Artificial Intelligence	L	P	C
		3	0	3

Course Objectives:

• Understand and analyze the importance and basic concepts of artificial intelligence and the use of agents.

• Identify, explore the complex problem-solving strategies and approaches.

• Analyze the concepts of basic concepts of neural networks and learning process.

• Explore and analyze the methodology used in machine learning.

Course Outcomes:

After completion of the course Students is able to

CO-1: Apply the basic concepts of artificial intelligence and the use of agents into the real-world scenario.

CO-2: Design and formulate complex problem solutions with the use of various searching techniques.

CO-3: Correlate the algorithmic approach of machine learning algorithms for a given case study.

CO-4: Analyze the phenomenon of neural networks and apply basic learning laws.

UNIT - I:

Introduction to AI – Foundations of AI – History of AI - Applications of AI, Intelligent Agents – Agents and Environments – Nature of Environments – Structure of Agents – Problem solving Agents – Problem formulation – Example Problems.

UNIT - II:

Searching Techniques – Uninformed Search Strategies – Breadth first search – Depth first search – Depth limited search - Bidirectional search – comparison – Search with partial information - Heuristic search – Greedy best first search – A* search – Memory bounded heuristic search - Heuristic functions - Local search- Hill climbing – Simulated annealing search - Local beam search, Genetic algorithms.

UNIT - III:

Constraint satisfaction problems – Backtracking search for CSP‟s - local search for constraint satisfaction problem. Adversarial search – Games - Minimax algorithm, Alpha beta pruning, cutting-off search.

Knowledge Representation and Reasoning: Propositional Logic, Rules of Inference, First Order Logic (FOL) Syntax, Semantics, Entailment.

UNIT - IV:

Classical Planning: Definition of Classical Planning, Algorithms for Planning with State Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.

Planning and Acting in the Real World: Time, Schedules, and Resources, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Multi agent Planning.

TEXTBOOKS:

1. Artificial Intelligence: A modern approach by Stuart Russell and Peter Norvig,3rd Edition, Prentice Hall, 2010.
2. Machine Learning by Tom M. Mitchell, M.C.Graw Hill Publications.
3. Neural networks A comprehensive foundation, Simon Haykin, Pearson Education, 2nd Edition, 2004

REFERENCES:

1. Artificial Intelligence by Elaine Rich & Kevin Knight,2nd Edition, TMH
2. Artificial Intelligence, A New Synthesis by Nils J. Nilsson, ELSEVIER
3. Artificial Neural Networks by Yegnanarayana.B, PHI

CD222 **Database Management Systems** **L P C**

3 0 3

Course Objectives

At the end of the course the students will understand

1. Fundamental concepts and architectures of database system
2. Features and design of conceptual and relational data models
3. Formal relational Languages and SQL to query, update, and manage a database
4. The concepts and protocols related to transaction processing, concurrency control and recovery

Course Outcomes

At the end of the course the students will be able to

1. Discuss the fundamental concepts and architecture of database systems.
2. Query the database using relational algebra and SQL.
3. Explain the concepts of relational data model and design database using normalization process.
4. Develop conceptual database schema for a given specification.
5. Describe the role of transaction processing, concurrency control and recovery in a multi user database system.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	1					1			2
CO2	3	2	2	2					1			2
CO3	3	2	2	2					2			2
CO4	2	2	1	1					1			2
CO5	3	3	3	2					2			3

CO-PSO Mapping

	PSO1	PSO2	PSO3
CO1	3	3	3
CO2	3	3	3
CO3	3	3	3
CO4	2	3	3

Course Content:

UNIT I

Introduction to Databases and Database Management System: Database system Applications, Advantages of DBMS over File System, Data Models, Instances and schema, View of Data, Database Languages –DDL, DML, DCL, Database Users and Administrator, Database System Architecture

Introduction to the Relational Model: Structure of RDBMS, Database Schema, Keys, Relational Query Languages, Relational Operations

UNIT II

Formal Relational Query Languages - The Relational Algebra and Relational Calculus

SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database, Join Expressions, Views, Transaction, Integrity Constraints, SQL Data Types and Schemas, Authorization

UNIT III

Database Design and the E-R Model - Overview of the Design Process, The Entity-Relationship Model, Constraints, Removing Redundant Attributes in Entity Sets, Entity-Relationship Diagrams, Reduction to Relational Schemas.

Relational Database Design - Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Functional-Dependency Theory, Algorithms for Decomposition, Decomposition Using Multivalued Dependencies, More Normal Forms, Database-Design Process.

UNIT IV

Transactions: Transaction Concept, A Simple Transaction Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels

Concurrency Control: Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Timestamp-Based Protocols, Validation-Based Protocols, Multiversion Schemes, Snapshot Isolation

Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management

Learning Resources:

Text Book:

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Sixth Edition, McGraw Hill Publishers

CD223	Operating Systems	L	P	C
		3	0	3

Course Objectives:

At the end of this course the students will understand

1. To introduce the structure and functions of the operating system
2. To provide the knowledge of how the operating system manages the resources
3. To expose the students to the issues related to executing multiple process in the system.

Course Outcomes:

1. Describe the fundamental concepts of an operating system functionality, and processes.
2. Apply the concepts of multithreading and IPC mechanisms.
3. Analyze the performance of CPU scheduling algorithms, page replacement algorithms, and disk scheduling algorithms.
4. Demonstrate the methods to solve critical section problem and deadlock handling in a system.
5. Differentiate the effectiveness and the hardware support required for contiguous, non-contiguous, and virtual memory management schemes.
6. Differentiate the file systems for applying different allocation and access techniques.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2	3	3	2									
CO3	3	3	2									
CO4	3	3	3									
CO5	3											
CO6	3											

CO-PSO Mapping

	PSO1	PSO2	PSO3
CO1	3		
CO2	3	2	
CO3	2	3	3
CO4	3	2	3
CO5	3		
CO6	2		3

Course Content:

UNIT I

Introduction: What Operating Systems Do, Operating-System Operations, Resource Management, Security and Protection, Virtualization, Distributed Systems, Kernel Data Structures.

Operating System Structures: Operating-System Services, User and Operating-System Interface, System Calls, Operating-System Structure.

Processes: Process Concept, Process Scheduling, Operations on Processes, inter process Communication, IPC in shared-memory Systems, IPC in Message-passing Systems.

UNIT II

Threads and Concurrency: Overview, Multicore Programming, Multithreading Models, Implicit Threading, Threading Issues.

CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Real-Time CPU Scheduling.

Synchronization: Background, The Critical-Section Problem, Peterson „solution, Hardware support for Synchronization, Mutex Locks, Semaphores, Monitors. Classic Problems of Synchronization.

UNIT III

Dead Locks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

Main Memory: Background, Contiguous Memory Allocation, Paging, Structure of the Page Table, Swapping.

Virtual-Memory: Background, Demand Paging, Page Replacement, allocation of frames, Thrashing - Memory Compression, Other considerations.

UNIT IV

Mass-Storage Structure: Overview of Mass-Storage Structure, HDD Scheduling.

Files System Interface: File Concept, Access Methods, Directory Structure, Protection, Memory –mapped files.

File-Systems Implementation: File-System Structure, File-System operations, Directory Implementation, Allocation Methods, and Free-Space Management.

Learning Resources:

Text Book(s):

1. Operating System Concepts-Abraham Silberchatz, Peter B Galvin, Greg Gange Tenth Edition, WILEY.

Reference Books:

1. Operating Systems, Internal and Design Principles, Stallings, 8th Edition-2015, Pearson education/PHI.
2. Operating system, A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tenenbaum 4th Edition Pearson/PHI.
4. An Introduction to Operating Systems, Concepts and Practice, 4th Edition, PHI, 2013-Pramod Chandra P. Bhatt.
5. Operating Systems- A concept based approach –DM Dhamdhere -3rd Edition TMH.

CD224	Software Engineering	L	P	C
		3	0	3

Course Objectives:

At the end of the course, the student will understand and

1. Acquire knowledge on the principles and process models for software development.
2. Explain the specific requirements for a given software project
3. Acquire knowledge on design concepts and user interface principles for Software development
4. Examine various testing techniques and metrics applicable to a Software project

Course Outcomes:

After successful completion of the course, students will be able to:

1. Describe the software engineering process model required to create a software system.
2. Discuss the software requirements and analyze a model for a software project.
3. Design and specify software components for real-world problems.
4. Evaluate various software testing techniques and metrics.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2										
CO2	3	3	2									2
CO3	2	3	3			3					2	3
CO4	3	3	3			3					3	3

CO-PSO Mapping

	PSO1	PSO2	PSO3
CO1			2
CO2			2
CO3		2	3
CO4		3	3

Course Content:

UNIT-I

Software and Software engineering: The Nature of Software, Defining Software, Software Application Domains, Legacy Software, The software Process.

The Software Process: Process Models: A Generic Process Model, defining a Framework Activity, identifying a task set, Process Assessment and Improvement, Prescriptive Process Models: The waterfall model, Prototyping Process model, Evolutionary process model, The Unified Process.

Agile Development: What Is Agility? What Is an Agile Process? Scrum Other Agile Process Models, Scrum, Other Agile Frameworks- The XP Framework.

UNIT-II

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Requirements gathering, developing use cases, Building the Analysis Model, Negotiating Requirements, Requirements monitoring, Validating Requirements.

Requirements Modelling: Requirements Analysis, Scenario-Based Modeling, Class-Based Modeling, Functional Modelling, Behavioural Modelling.

Design Concepts: Design within the Context of Software Engineering, the Design Process, Design Concepts, the Design Model.

UNIT-III

Architectural Design: Software Architecture, Agility and Architecture, Architectural Styles, Architectural Design, Assessing Alternative Architectural Designs, Architectural Reviews.

Modeling Component-Level Design: What Is a Component? Designing Class-Based Components, Conducting Component Level Design.

User Experience Design: User Experience Design Elements, The Golden Rules, User Interface Analysis and Design, Interface Analysis and Design Models, The process.

UNIT-IV

Software Testing –Component Level: A Strategic Approach to Software Testing, Planning and Record keeping, Test case design, White box testing, Black-Box-Testing.

Software-Testing Integration level: Software Testing Fundamentals, Integration testing, Validation Testing, Testing Patterns.

Software Metrics and Analytics: Software Measurement, Software Analytics, Product Metrics, Metrics for Testing, Metrics for maintenance, Process and Project Metrics, Metrics for Quality.

Learning Resources:

Textbook(s):

1. Roger Pressman and Bruce Maxim "Software Engineering- A Practitioner's Approach", 9th edition, Tata McGraw-Hill International.

Reference Books:

1. Ian Somerville, Software Engineering. 6 ed, Pearson Education.
2. Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli, Fundamentals of Software Engineering.2 ed, PHI.
3. RajibMall, Fundamentals of Software Engineering. 2 ed, PHI.

Web Resources:

1. http://nptel.ac.in/courses/106101061/2
2. http://nptel.ac.in/courses/106101061/5

CD225	**Design & Analysis of Algorithms**	L	P	C
		3	0	3

Course Objectives:

The main objectives of this course are

1. Instruct performance analysis of an algorithm.
2. Illustrate algorithm design Strategies.
3. Demonstrate pattern matching algorithms
4. Impart knowledge on P, NP and NP-complete and NP-hard class of problems.

Course Outcomes:

After completion of the course, the students will be able to

CO1. Analyze the performance of algorithms
CO2. Apply algorithm design techniques to solve real world problem
CO3. Make use of string matching algorithms to solve complex problems
CO4. Solve P class and NP class problems

Course Content:

UNIT – I **CO1** **14 Periods**

Introduction - Algorithm Definition, Algorithm Specification – Pseudocode Conventions, Recursive Algorithms, Performance Analysis- space Complexity, Time Complexity, Asymptotic Notations, Practical Complexities and Performance Measurement.

Divide and Conquer: General Method, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Strassens Matrix Multiplication.

UNIT – II **CO2** **14 Periods**

Greedy Programming: General Method, Knapsack problem, Job Sequencing with Dead Lines, Minimum-cost Spanning Tree - Prim's and Kruskal's algorithms, Single-Source Shortest Paths- Dijkstra's Algorithm.

Basic Traversal & Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected Components and Spanning Trees, Bi-Connected Components and DFS.

UNIT – III **CO3** **14 Periods**

Dynamic Programming: General Method, Multi Stage Graph, All Pairs Shortest Paths, Single Source Shortest Paths-general Weights, Optimal Binary Search Trees, String Editing, 0/1 Knapsack, Traveling Salesman Problem.

Back Tracking: General Method, 8-queen problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles.

UNIT – IV **CO4** **14 Periods**

Branch and Bound: Control Abstraction for LC Search, 15 Puzzle Problem, Bounding, FIFO branch and bound, LC branch and bound, 0/1 Knapsack problem, Travelling Salesman Problem.

NP-hard and NP-Complete Problems: Basic Concepts, Non-Deterministic Algorithms, The classes NP-hard and NP-Complete, Cooks Theorem, NP-Hard Graph Problem, Click Decision Problem, Node Cover Decision Problem. (Theorem Proofs Excluded)

Learning Resources:

Text Book:

1. E.Horowitz, S. Sahni and S.Rajsekaran, "Fundamentals of Computer Algorithms", Galgotia Publication.

Reference Books:

1. T. H. Cormen, Leiserson, Rivest and Stein, "Introduction of Computer Algorithm", PHI. (Unit IV).
2. Sara Basse, A.V. Gelder, "Computer Algorithms", Addison Wesley.

CD261	Database Management Systems Lab	L	P	C
		0	3	1.5

Course Objectives:

At the end of the course the students will understand

1. Syntax and usage of DDL, DML, DCL, and TCL statements, asserting database integrity constraints during database creation.
2. Semantics of SQL for implementing the user queries on a relational database.
3. Block structured PL / SQL programming concepts.

Course Outcomes:

At the end of the course the students will be able to

1. Define, manipulate and control data using Structured Query Language (SQL).
2. Identify various database integrity constraints during database creation.
3. Construct SQL statements for satisfying end user queries by utilizing functions, set operations, joins, and sub queries.
4. Develop various applications using various PL/SQL data object like Database cursors, Functions, Stored Procedures, Packages, and Triggers.

COs - POs MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	1	2				2			2
CO2	2	2	2	2	2				2			2
CO3	3	3	3	3					2			2
CO4	3	3	3	3					2			2

Week 1

Practice DDL and DML statements for creating a sample database without integrity constraints.

Week 2

Practice DDL and DML statements for refining a sample database including integrity constraints.

Week 3

Query the sample database using simple select statements retrieving:

1. Small-large number of attributes
2. Distinct output values
3. By Renaming attributes
4. Computed attributes
5. By using Simple-complex conditions (AND, OR, NOT)
6. By using Partial Matching operators (LIKE, %, _, *,?)
7. Sorted records
8. By checking for Nulls

Week 4-6
Query the sample database using joins, nested queries, aggregate functions and set oriented operations
Week 7 Query the sample database using built-in single row functions
Week 8 Implement PL/SQL named and unnamed blocks
Week 9 Implement PL/SQL Implicit and Explicit Cursors
Week 10 Implement PL/SQL pre-defined and user defined exceptions
Week 11 Implement PL/SQL stored procedures, functions and packages
Week 12 Implement PL/SQL database triggers

CD262	Operating Systems & Software Engineering Lab	L	P	C
		0	3	1.5

List of Experiments: (Operating Systems Lab)

1. Simulate the following non-preemptive CPU scheduling algorithms to find turn around time and waiting time: a.FCFS b.SJF c.Round Robin
2. Simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories-system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.
3. Simulate the following file allocation strategies: a.Sequential b.Indexed c. Linked
4. Simulate the MVT and MFT memory management techniques.
5. Simulate the following contiguous memory allocation techniques:
   a.Worst-fit b.Best-fit c.First-fit
6. Simulate paging technique of memory management.
7. Simulate Bankers algorithm for the purpose of deadlock avoidance.
8. Simulate page replacement algorithms: a.FIFO b.SC c.NRU d.LRU
9. Simulate disk scheduling algorithms: a.FCFS b.SCAN c.C-SCAN
10. Simulate producer-consumer problem using semaphores.
11. Basics of UNIX commands.
12. Shell Programming

List of Experiments: (Software Engineering Lab)

Lab cycle 1:

Development of requirements specification, function oriented design using SA/SD.

Lab cycle 2:

Object-oriented design using UML.

1. ANALYSIS- SRS documentation
2. USECASEVIEW
   i. Construction of use case model
   ii. Building a Business Process model using UML activity diagram.
3. LOGICAL VIEW- Construction of UML static class diagram.

Sample information systems for implementation:

1. Course registration system
2. ATM services
3. Advertising agency management system
4. Online shopping
5. Library management system

CD263	Design and Analysis of Algorithms Lab	L	P	C
		-	3	1.5

Course Objectives:

Students will gain practical exposure with designing and implementing different algorithms design strategies such as Divide and Conquer, Greedy method, Dynamic Programming, Backtracking and Branch and Bound.

Course Outcomes:

After the successful completion of the course, students will be able to

1. Device an algorithm with an appropriate design strategy.
2. Implement the algorithm in a high-level language
3. Analyze the performance of algorithms using language features.

Experiments:

1. Find the min-max of the list of elements using DAC.
2. Find the kth smallest element using DAC.
3. Calculate the optimal profit of a Knapsack using the Greedy method.
4. Determine the path length from a source vertex to the other vertices in a given graph. (Dijkstra's algorithm)
5. Construct a minimum cost spanning tree for the given graph. (Kruskal's algorithm)
6. Determine the shortest path in a multi-stage graph using the forward and backward approach.
7. Find the Shortest path from any node to any other node (All-pairs Shortest path) within a graph.
8. Construct spanning trees using DFS and BFS graph traversals.
9. Find the bi-connected components in a graph
10. Find the non-attacking positions of Queens in a given chess board using the backtracking Technique.
11. Color the nodes in a given graph such that no two adjacent can have the same color using the backtracking Technique.
12. Calculate the optimal profit of a Knapsack using Branch and Bound Technique.

CDSL2	Scripting Languages	L	P	C
		3	0	2

Prerequisites:

- A course on "Computer Programming and Data Structures."

Course Objectives:

The goal of the course is to study:

- Modify built-in shell variables and create and use user-defined shell variables.
- Create structured shell programming which accept and use positional parameters and exported variables.
- Use shell flow control and conditional branching constructs (while, for, case, if, etc.)
- Understand and make effective use of Linux utilities and Shell scripting language (bash) to solve Problems.
- Basic introduction to programming using Perl.
- Knowledge of CGI scripts.
- To understand basic concepts of PHP language and developing web applications.

Course Outcomes:

Upon learning the course, the student will have the:

- Understand the basic commands of linux operating system and can write shell scripts.
- Understand basics of Perl.
- Understand list arrays and hash.
- Develop web applications using PHP.

UNIT-I

Introduction to Unix: Unix utilities – process utilities, disk utilities, networking commands, text processing utilities and backup utilities. Introduction to unix file system, vi editor, file handling utilities, security by file permissions.

Shell Programming-Shell Variables-The Export Command-The Profile File a Script Run During Starting-The First Shell Script-The read Command-Positional parameters-The $? Variable knowing the exit Status-More about the Set Command-The Exit Command-Branching Control Structures-Loop Control Structures-The Continue and Break Statement-The Expr Command: Performing Integer Arithmetic-Real Arithmetic in Shell Programs- The here Document (<<)-The Sleep Command-Debugging Scripts-The Script Command-The Eval Command-The Exec Command.

UNIT – II

Introduction to PERL and Scripting, Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the

universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

UNIT-III

JavaScript: Introduction to Scripting, Control Statements-I, Control Statements-II, Functions and Arrays.

UNIT – IV

PHP Basics – Features, Embedding PHP Code in your Web pages, outputting the data to the browser, Data types, Variables, Constants, expressions, string interpolation, control Structures. Function, Creating a Function, Function Libraries, Arrays, Strings and Regular Expressions.

Text Books:

1. Unix for programmers and users, Graham Glass, King Ables, 3rd edition, Pearson education. [UNIT-I]
2. Introduction to Unix Shell Programming by M.G.Venkateshmurthy, Pearson. [UNIT-I]
3. Perl by Example, E.Quigley, Pearson Education. [UNIT-II]
4. Harvey M. Deitel and Paul J.Deitel, "Internet & World Wide Web How to Program", 4/e, Pearson Education. [UNIT-III]
5. PHP: The Complete Reference, Steven Holzner, TATA McGraw Hill, 2013. [UNIT-IV]

Reference Books:

1. Your Unix the ultimate guide, Sumitabha Das, TMH. 2nd Edition.
2. Unix and shell programming by B.M. Harwani, OXFORD university press.
3. The Unix programming Environment by Brain W. Kernighan & Rob Pike, Pearson.
4. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O"Reilly, SPD.
5. Perl Power, J.P. Flynt, Cengage Learning.
6. The World of Scripting Languages, David Barron, Wiley Publications.

CDMC4	Ethics &Human Values	L	P	C
		2	0	-

Course Objectives:

At the end of the course the students will understand

- To create awareness to specific set of morals, values and ethics the professional must know and abide by, including work ethics, integrity and commitment etc.
- To realize the importance of moral autonomy, professional ideals and Ethical theories
- To study safety/risk aspects, welfare of the public and about employee rights
- Know about the global issues and code of ethics of professional bodies

Course Outcomes:

At the end of the course the students will be able to

- Have basic understanding of how a prospective engineers should behave in his chosen field and society.
- Realize the importance of moral autonomy, professional ideals and Ethical theories.
- Know about the safety/risk, welfare of the public and employee rights
- Gain exposure to global issues and codes of some professional bodies

UNIT I

Human Values: Morals, Values and Ethics - Integrity- Work Ethics- Service Learning – Civic Virtue Respect for Others-Living Peacefully-Caring-Sharing-Honesty-Courage–Valuing Time–Co-Operation-Commitment-Empathy-Self-Confidence–Stress Management-Character-Spirituality.

UNIT II

Engineering Ethics: Senses of Engineering Ethics-Variety of Moral Issues-Types of Inquiry-Moral Dilemmas-Moral Autonomy-Kohlberg's Theory-Gillian-s Theory -Consensus and Controversy.

Professions and Professionalism: The nature and characteristics of Professions, Professionalism, the foundation and norms of Professional ethics, the need for separate code of conduct for Professionals, Professional Rights, Theories about Right Action, Uses of Ethical Theories. Case studies like The Space Shuttle Challenger, Bhopal gas tragedy, Chernobyl disaster etc.

UNIT III

Engineering as Social Experimentation: Engineering as Experimentation-Engineers as Responsible Experimenters Safety.

Responsibilities and Rights: Safety and Risk-Assessment of Safety and Risk, Risk Benefit Analysis and Reducing Risk. Collegiality and Loyalty – Respect for Authority –Collective Bargaining - Confidentiality –Conflicts of Interest-Occupational Crime-Employee Rights–Intellectual Property Rights (IPR)-Discrimination.

UNIT IV

Multinational Corporations-Environmental Ethics-Computer Ethics-Business ethics-Engineers As Managers- Consulting Engineers - Engineers As Expert Witnesses and Advisors – Codes Of Ethics –Sample Code Of Ethics Like ASME,A SCE, IEEE, Institution of Engineers(India), Indian Institute of Materials Management Etc.,

Learning Resources:

Text Books:

1. Mike martin and Ronald Schinzinger, "Ethics in Engineering"McGraw-Hill,NewYork1996
2. Govinda rajan M, NatarajanS, Senthil KumarV.S., "Engineering Ethics", PHI, New Delhi
3. Bayles.M.D,Professional ethics,California, Wards worth publishing company,1981
4. Koehn.D, The ground of Professional Ethics,Routledges,1995

Reference Books:

1. Charles D,Fleddermann,"EngineeringEthics",Pearson/PHI,NewJersey2004(IndianReprint)
2. Charles E Harris, MichaelS.Protchard and MichaelJRabins,"EngineeringEthics-ConceptsandCases"WadsworthThompsonLearning,UnitedStates,2000(IndianReprintnowavailable)
3. John R Boatright, "Ethics and the conduct of business" Pearson, New Delhi,2003.
4. Edmund, G.Seebauer and Robert L Barry, "Fundamentals of Ethics for Scientists and Engineers "Oxford University Press, Oxford, 2001.