BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING / COMPUTER SCIENCE AND INFORMATION TECHNOLOGY / INFORMATION TECHNOLOGY / COMPUTER SCIENCE AND TECHNOLOGY

THIRD YEAR (FIFTH SEMESTER)

W.E.F. ADMISSION BATCH 2023-24

Sl. No.	Category	Course Code	Course	Contact Hrs. L-T-P	Credit	University Marks	Internal Evaluation
Subject (Theory)							
1	PC	CSPC3001	Theory of Computation	3-0-0	3	100	50
2	PC	CSPC3002	Operating Systems	3-0-0	3	100	50
3	PC	CSPC3003	Artificial Intelligence and Machine Learning	3-0-0	3	100	50
4	PE	CSPE3001	Computational Intelligence	3-0-0	3	100	50
		CSPE3002	Pattern Recognition				
		ECPC3001	Microprocessor & Microcontroller				
		CSPE3004	Distributed Systems				
		CSPE3005	Cryptography and Network Security				
		CSPE3006	Object Oriented Analysis and Design				
		CSPE3007	Web Technology				
5	HS	HSHS3002	Entrepreneurship Development	3-0-0	2	100	50
		HSHS3003	Professional Ethics				
6	MC	MCMC3001	Environmental Engineering	3-0-0	2	100	50
		MCMC3002	Industrial Safety Engineering				
Subject (Sessional / Practical)							
7	PC	CSPC3201	Theory of Computation Laboratory	0-0-3	1.5	-	100
8	PC	CSPC3202	Operating Systems Laboratory	0-0-3	1.5	-	100
9	PC	CSPC3203	Machine Learning Laboratory	0-0-3	1.5	-	100
10	PSI	CSPS3201	Seminar on SIRE - I	0-0-3	1.5	-	100
			Total	**18-0-12**	**22**	**600**	**700**

Click here to view/download the syllabus of the subjects.

CSPC3001 THEORY OF COMPUTATION (3-0-0)

Course Objectives

The course aims to:

1. Introduce the mathematical foundations of computation through formal languages, grammars, automata, and Turing machines.
2. Develop students' ability to design abstract models of computation and analyze their capabilities and limitations.
3. Understand the concepts of decidability, reducibility, and computational complexity.
4. Foster mathematical rigor in designing and proving properties of computational models.
5. Explore the Chomsky hierarchy and the relationships between various computational classes.

Module-I: Fundamentals and Finite Automata (08 Hours)

Basic definitions: alphabets, strings, languages, and operations on languages. Introduction to finite automata: deterministic finite automata (DFA), nondeterministic finite automata (NFA), and transition diagrams. Language recognition, equivalence between DFA and NFA, and conversion of NFA to DFA. Epsilon (ε) transitions in NFA and their elimination. Minimization of finite state machines. Moore and Mealy machines, and conversion between them.

Module-II: Regular Languages and Expressions (08 Hours)

Regular expressions and identity rules. Construction of finite automata from regular expressions and vice versa. Regular grammars: right linear and left linear forms. Conversion between different regular representations (grammar, FA, and regex). Pumping lemma for regular languages and its application. Closure properties of regular languages.

Module III: Context-Free Grammars and Pushdown Automata (08 Hours)

Definition and examples of context-free grammars (CFGs), derivation trees, leftmost and rightmost derivations, ambiguity in CFGs. Simplification of CFGs, Chomsky and Greibach normal forms. Pumping lemma for context-free languages and its applications. Pushdown automata (PDA): definition, model, language acceptance by final state and empty stack. Equivalence between CFG and PDA. Introduction to deterministic and nondeterministic PDA, and comparison.

Module IV: Turing Machines and Computability (8 Hours)

Turing Machine (TM): formal definition, configurations, and design of TMs. Variants of TMs and their equivalence. Linear bounded automata (LBA) and context-sensitive languages. Computable functions, recursively enumerable and recursive languages. Church–Turing thesis. Decidability and undecidability, examples of undecidable problems. Reductions and mapping reducibility.

Module V: Complexity Theory and Chomsky Hierarchy (08 Hours)

Chomsky hierarchy: classification of languages and corresponding machines. Introduction to complexity theory: efficiency of computation, time and space complexity. Complexity classes P and NP, NP-completeness, and the P vs NP problem. Introduction to polynomial-time reductions and basic NP-complete problems. Language families and their relationships.

Course Outcomes

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

CO1. Explain the fundamental concepts of automata theory and formal languages.
CO2. Design finite automata, pushdown automata, and Turing machines for formal language recognition.
CO3. Apply regular expressions, context-free grammars, and normal forms in language design.
CO4. Analyze decidability and computability of languages and problems.
CO5. Classify languages and problems using the Chomsky hierarchy and complexity classes.
CO6. Prove formal properties of computational models using pumping lemmas and reduction techniques.

Text books

1. Introduction to Automata Theory, Languages, and Computation – Hopcroft H.E. and Ullman J.D., Pearson Education.
2. An Introduction to Formal Languages and Automata – Peter Linz, Narosa Publishing.
3. Introduction to the Theory of Computation – Michael Sipser, 2nd Edition, Thomson.

Reference Books

1. Introduction to Computer Theory – Daniel I.A. Cohen, John Wiley.
2. Introduction to Languages and the Theory of Computation – John C. Martin, TMH.
3. Elements of the Theory of Computation – Lewis H.P. & Papadimitriou C.H., Pearson.
4. Theory of Computer Science (Automata, Languages and Computation) – Mishra and Chandrashekaran, PHI.

CSPC3201 THEORY OF COMPUTATION LABORATORY (0-0-3)

Course Objectives:

- To provide hands-on experience with concepts of automata theory, formal languages, and Turing machines.
- To simulate and experiment with finite automata, context-free grammars, pushdown automata, and Turing machines.
- To reinforce understanding of key computation concepts such as decidability, complexity, and algorithm efficiency through practical tasks.

List of Experiments:

1. Introduction to Automata Simulation Tools
   - Installation and familiarization with software tools like JFLAP for automata simulation.
   - Basic operations: creating and running simple finite automata.
2. Design and Simulation of Deterministic Finite Automata (DFA)
   - Construct DFA for given regular languages.
   - Validate the DFA by inputting sample strings to check acceptance/rejection.
3. Design and Simulation of Non-Deterministic Finite Automata (NFA)
   - Construct NFA for various regular languages.
   - Convert the designed NFA into an equivalent DFA and compare their performance.
4. NFA with ε-transitions
   - Design an NFA with ε-transitions for specific regular expressions.
   - Convert this NFA to a DFA.
5. Finite Automata with Output (Moore and Mealy Machines)
   - Design Moore and Mealy machines for particular output requirements.
   - Convert a Mealy machine to a Moore machine and vice-versa.
6. Context-Free Grammar (CFG) Design
   - Design CFGs for different context-free languages.
   - Parse strings using the designed CFG and generate parse trees.
7. PDA (Pushdown Automata) Simulation
   - Design a PDA for specific context-free languages.
   - Simulate PDA for both acceptance by final state and acceptance by empty stack.
8. Turing Machine Design
   - Design and simulate a Turing machine for basic mathematical operations (e.g., addition, subtraction).
   - Experiment with various inputs to test Turing machine behavior.

Tools/Software:

- JFLAP or any equivalent automata simulation tool for performing practicals.

Course Outcome:

Upon completing the laboratory course, students will be able to:

- Design and simulate finite automata, pushdown automata, and Turing machines.
- Apply theoretical concepts of automata theory in practical scenarios.
- validate theoretical concepts like regular expressions, context-free grammars, and pumping lemmas through practical experiments
- Develop problem-solving skills using computation models and improve their understanding of algorithm efficiency and complexity.
- simulate Turing Machines to understand their functioning and apply them to solve computational problems.

CSPC3002 OPERATING SYSTEMS (3-0-0)

Course Objectives:

- To provide knowledge about the services rendered by operating systems
- To provide a detailed discussion of the various memory management techniques
- To discuss the various file-system design and implementation issues
- To discuss how the protection domains help to achieve security in a system

Module-I (10hours)

Introduction to OS- About an OS, Simple batch system, multiprogramming and time-sharing system.
Operating system structure- operating system services, system components, protection system, OS service, System call
Process Management- Process Concepts, Process Scheduling, Operation on process, IPC, multi-threading models, Threading issues, Process Scheduling Algorithms.

Module-II (10hours)

Process Coordination & Synchronization- Critical Section Problem, Peterson's solution, Synchronization hardware, Semaphores, Classical problems of Synchronization.
Dead-locks – System model, Deadlock Characterization, Methods for handling Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

Module-III (08 hours)

Memory Management-Memory Management Strategies, Logical v/s Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation
Virtual Memory-Background, Demand Paging, Page Replacement Algorithm, Allocation of Frame, Thrashing, Demand segmentation

Module-IV (08 hours)

Storage Management- File System concept, Access Methods, File System Structure, implementation, Efficiency & Performance, Recovery, Overview of Storage Structure, Disk Structure, Disk Scheduling, Disk Management, Swap Space Management, I/O System overview, I/O Hardware, application I/O interface, Kernel I/O subsystem, Transforming I/O request to H/W operation.

Module-V (04hours)

Distributed systems - Distributed file systems, Distributed operating sytems, Real time systems.

Outcomes

CO1: Ability to comprehend the techniques used to implement the process manager.
CO2: Ability to comprehend virtual memory abstractions in operating systems.
CO3: Ability to design and develop file system interfaces, etc.
CO4: Ability to comprehend the techniques used to implement the process manager.
CO5: Ability to comprehend virtual memory abstractions in operating systems.

Books:

1. Silberschatz, Galvin, Gagne, "Operating System Concepts", John Wiley and Sons, 10th edition.
2. Stallings, "Operating Systems –Internals and Design Principles", 9/E, Pearson Publications, 2018.
3. Andrew S. Tanenbaum, "Modern Operating Systems", 4/E, Pearson Publications.

CSPC3202 OPERATING SYSTEMS LABORATORY (0-0-3)

Course Objectives

- Understand the practical aspects of process management, memory management, file systems, and scheduling.
- Implement core operating system functionalities such as scheduling, synchronization, and deadlock handling.
- Gain hands-on experience in using Linux commands and shell scripting.
- Simulate key OS components through programming.

List of Experiments:

1. Installation of operating system.
2. Linux Administrative commands.
3. UNIX Shell Programming.
4. Process Management using System Calls: fork(), exit(), getpid(), wait(), close().
5. Implementation of Dinning Philosopher problem / Cigarette Smoker problem / Sleeping barber problem.
6. Simulation of CPU Scheduling Algorithms. (FCFS, RR, SJF).
7. Simulation of Banker's Algorithm for Deadlock Avoidance, Prevention.
8. Program for FIFO, LRU, and OPTIMAL page replacement algorithm.
9. Write C program to implement Threading
10. Implement the paging scheme using C program.
11. Write C programs to implement the following Memory Allocation Methods
    a. First Fit b. Worst Fit c. Best Fit
12. Android Programming for mobile application.

Course Outcomes

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

1. Demonstrate proficiency in Unix/Linux commands and shell scripting.
2. Implement process scheduling and synchronization algorithms.
3. Solve concurrency-related problems using semaphores and threads.
4. Understand memory management techniques and simulate page replacement.
5. Develop basic system programs for OS operations such as file handling and process communication.
6. Apply operating system principles in a simulated or real-world environment.

CSPC3003 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (3-0-0)

Course Objectives:

- To learn the concepts of Artificial Intelligence
- To learn the methods of solving problems using Artificial Intelligence
- To introduce the concepts of Expert Systems and machine learning

Module-I: (10 hours)

Introduction To AI –The Foundations of Artificial Intelligence; Agents and Environments, Intelligent Agent , Good Behaviour-The Concept of Rationality, the Nature of Environments, Structure of Agents.

Problem-solving and searching in AI- Iterative search, Uninformed search (Breadth-first search, Depth-first search, Uniform cost search), Informed (Heuristic) Search Strategies (Greedy best-first search, A* Search), Constraint Satisfaction Problem (CSP).

Module-II: (10 hours)

Adversarial search – Games, The Mini-Max algorithm, Alpha-Beta Pruning.

Knowledge Representations & Reasoning– logical agents, Knowledge-Based Agents, Logic, Reasoning Patterns in Propositional Logic & First-Order Logic, Resolution Using Propositional logic & First-Order Logic. Unification and Lifting, Forward Chaining, Backward Chaining,

Module-III: (08 hours)

Uncertainty – Reasoning with uncertainty (Probabilistic Reasoning) Bayes' rule, Bayesian Network Representation, Markov Models, Independence and Inference.

Expert Systems: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition

Module-IV: (06 hours)

Learning methods – Statistical Learning, Rote Learning, Learning by Taking Advice, Learning in Problem-solving, Learning from Examples: Induction, Explanation-based Learning.

Module-V: (08 hours)

Machine learning: Supervised learning, Unsupervised learning, Reinforcement learning

Machine Learning Paradigms- Regression (Linear, Logistic), Classification (decision trees, KNN, support vector machine, Naïve Bayes Classifier), Model Evaluation Metrics: Overfitting and underfitting, Clustering, Dimensionality Reduction, Neural Network basics, Ensemble learning.)

Course Outcomes

- Ability to comprehend AI & ES to analyze and map real world activities to digital world
- Ability to identify problems that are amenably solved by AI methods
- Ability to design and carry out an empirical evaluation of different AI algorithms

After completing this course, the student must demonstrate the knowledge and ability to:

CO1: Reason about the state-space search algorithm to use under different problem specific conditions.
CO2: Get deep insight of AI and its problem Solving techniques.
CO3: Implement probabilistic solutions for decision making such as Hidden Markov Models, Bayes' Networks, etc.
CO4: Learn and implement basic supervised methods like Decision Trees, Nearest Neighbours, Perceptron, Linear regression, Logistic regression, SVM and Ensemble Techniques.
CO5: Gain an understanding of basic unsupervised methods like Clustering.

Books:

1. Elaine Rich, Kevin Knight, & Shivashankar B Nair, Artificial Intelligence, McGraw Hill,3rd ed.,2009
2. Stuart Russell, Peter Norvig, Artificial Intelligence -A Modern Approach, 4/e, Pearson, 2003.
3. Nils J Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publications,2000.
4. Introduction to Artificial Intelligence & Expert Systems, Dan W Patterson, PHI.,2010
5. S Kaushik, Artificial Intelligence, Cengage Learning, 1st ed.2011
6. Aurlien Geron, Hands on Machine Learning with Scikit & TensorFlow- Concepts ,Tools & Techniques To Build Intelligent Systems

CSPC3203 MACHINE LEARNING LABORATORY (0-0-3)

Course Objectives:

- Understand the basics of machine learning and its applications
- Implement machine learning algorithms using popular libraries (e.g.Pandas, Matplotlib, scikit-learn, TensorFlow, PyTorch)
- Apply machine learning techniques to real-world problems
- Develop skills in data preprocessing, model evaluation, and hyperparameter tuning

Lab Assignments:

Introduction to Python Programming

1. Install Python and set up Anaconda.
   Write basic Python scripts including loops, conditional statements, and functions.
2. Introduction to Machine Learning Libraries using Python:
3. NumPy: Perform matrix operations, loops, and conditions.
   Matplotlib: Create and customize plots.
   Pandas: Load, explore, and summarize datasets.
4. Find the mean, median, mode, variance and standard deviation of a list.
5. Overview of scikit-learn, TensorFlow, and PyTorch
6. Implement the linear regression algorithm.
7. Implement the logistic regression algorithm.
8. Implement the K-nearest neighbour algorithm.
9. Implement the decision tree algorithm.
10. Implement the random forest algorithm.
11. Implement the support vector machine algorithm.
12. Implement the principal component analysis algorithm.
13. Implement the k – means clustering algorithm.

Real-World Applications:

- Image classification
- Text classification
- Regression problems (e.g., housing prices)

CSPE3007 WEB TECHNOLOGY (3-0-0)

Course Objectives:

- Illustrate the Semantic Structure of HTML and CSS.
- Compose forms and tables using HTML and CSS.
- Design Client-Side programs using JavaScript and Server-Side programs using PHP.
- Infer Object Oriented Programming capabilities of PHP.
- To create more interactive and responsive web applications by allowing web pages to update content without reloading the entire page

Module-I: (08 Hrs)

Introduction to HTML

Web Essentials: Clients, Servers and Communication:

The Internet – Basic Internet protocols – The WWW, HTTP request message – response message, web clients, web servers.

Introduction to HTML, what is HTML and Where did it come from? HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility.

Module-II: (08 Hrs)

Introduction to CSS

What is CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling.

Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.

Module-III: (08 Hrs)

JavaScript: Client-Side Scripting

What is JavaScript and What can it do? JavaScript Design Principles, where does JavaScript Go? Syntax, JavaScript Objects, The Document Object Model (DOM), JavaScript Events, Forms, Introduction to Server-Side Development with PHP, what is Server-Side Development, A Web Server's Responsibilities, Quick Tour of PHP, Program Control, Functions.

Module-IV: (08 Hrs)

Introduction to PHP

Declaring variables, data types, arrays, strings, operations, expressions, control structures, functions, reading data from web form controls like Text Boxes, radio buttons, lists etc., Handling File Uploads, connecting to database (My SQL as reference), executing simple queries, handling results, Handling sessions and cookies.

File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

Module-V: (08 Hrs)

Introduction to AJAX

Exploring different web technologies, creating a simple AJAX application, interacting with the Web Server Using the XML Http Request Object, Create an XML Http Request Object, Interact with the Web Server. Differentiating AJAX and Non-AJAX application.

Working with PHP and AJAX: Introduction, Process Client Requests, Accessing Files Using PHP, Implementing Security and Accessibility in AJAX applications: Introduction, Secure AJAX Applications, and Accessible Rich Internet Applications.

Course Outcomes:

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

- Demonstrate the ability to construct web pages using HTML5, CSS and CSS frameworks.
- Develop Client-Side Scripts using JavaScript.
- Develop Server-Side Scripts using PHP to generate and display the contents dynamically.
- Appraise the principles of object-oriented development using PHP.
- Build dynamic, interactive and secure web applications using AJAX and PHP.

Text Books:

1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 1stEdition, Pearson Education India.
2. Zak Ruvalcaba Anne Boehm, "Murach's HTML5 and CSS3", 3rdEdition, Murachs/Shroff Publishers & Distributors Pvt Ltd, 2016.

Reference Books:

1. Robin Nixon, "Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5", 4thEdition, O'Reilly Publications, 2015.
2. Nicholas C Zakas, "Professional JavaScript for Web Developers", 3rd Edition, Wrox/Wiley India, 2012.

HSHS3003 PROFESSIONAL ETHICS (3-0-0)

Course Objectives :

This course aims to develop students' understanding of ethical principles, moral reasoning, and professional responsibilities. It introduces ethical theories, value systems, and the role of ethics in engineering and global contexts. The course prepares students to identify, analyze, and resolve ethical dilemmas in professional and societal scenarios with integrity.

MODULE-I (6 Hours)

Introduction to Ethics:
Basic terms- Moral, Ethics, Ethical dilemma, Emotional intelligence
Moral development theories of Kohlberg and Piaget
View on ethics by Aristotle
Governing factors of an individual's value system
Personal and professional ethics

MODULE-II (6 Hours)

Profession and Professionalism:
Clarification of the concepts: Profession, Professional, Professionalism, Professional accountability, Professional risks, Profession and Craftsmanship, Conflict of interest, Distinguishing features of a professional, Role and responsibilities of professionals, Professionals' duties towards the organization and vice-a-versa, Ethical Theories: Various ethical theories and their application- Consequentialism, Deontology, Virtue theory, Rights Theory, Casuist theory
Ethical terms: Moral absolutism, Moral Relativism, Moral Pluralism etc.; Resolving Ethical Dilemma

MODULE-III (6 Hours)

Ethics in Engineering:
Purpose and concept of Engineering Ethics
Engineering as social experimentation
Types of inquiry
Issues in engineering ethics
Engineers' Responsibility and Safety:
Safety, Risk, Understanding the risk, Over estimating the risk, Risk-benefit analysis
Causes of an accident and identification of the preventive measures to be taken
Case Studies

MODULE-IV (6 Hours)

Global Ethical Issues:
Different ethical issues in business, environment, IT, Bioethics, Intellectual Property Rights (IPR), Research, Media, CSR etc.
Ethical Codes:
Meaning and the significance of ethical codes
The limitations of ethical codes.

Course Outcome

CO1: Define key ethical terms
CO2: Identify factors influencing value systems
CO3: Describe professional concepts
CO4: Explain the purpose of engineering ethics

BOOKS FOR REFERENCE:

1. R. Subramanian, "Professional Ethics", Oxford University Press
2. Mike W. Martin and Roland Schinzinger, "Ethics in Engineering", Tata McGraw Hill
3. Charles E Harris, Michael S Pritchard and Michael J Rabins, "Engineering Ethics - Concepts and Cases", Thompson Learning
4. Daniel Albuquerque, "Business Ethics", Oxford University Press
5. Edmund G. Seebauer and Robert L. Barry, "Fundamentals of Ethics", Oxford University Press

MCMC3002 INDUSTRIAL SAFETY ENGINEERING (3-0-0)

Course Objectives:

1. Students will be able to recognize and evaluate occupational safety and health hazards in the workplace, and to determine appropriate hazard controls following the hierarchy of controls.
2. Students will furthermore be able to analyze the effects of workplace exposures, injuries and illnesses, fatalities and the methods to prevent incidents using the hierarchy of controls, effective safety and health management systems and task-oriented training.

Course Outcomes:

By the end of this course, a student should:

CO1: Evaluate workplace to determine the existence of occupational safety and health hazards
CO2: Identify relevant regulatory and national consensus standards along with best practices that are applicable.
CO3: Select appropriate control methodologies based on the hierarchy of controls
CO4: Analyze injury and illness data for trends

Module-I: (07 hrs)

Industrial safety: Accident, causes, types, results and control, mechanical and electrical hazards, types, causes and preventive steps/procedure, describe salient points of factories act 1948 for health and safety, wash rooms, drinking water layouts, light, cleanliness, fire, guarding, pressure vessels, etc, Safety color codes. Fire prevention and firefighting, equipment and methods.

Module-II: (07 hrs)

Fundamentals of maintenance engineering: Definition and aim of maintenance engineering, Primary and secondary functions and responsibility of maintenance department, Types of maintenance, Types and applications of tools used for maintenance, Maintenance cost & its relation with replacement economy, Service life of equipment.

Module-III: (07 hrs)

Wear and Corrosion and their prevention: Wear- types, causes, effects, wear reduction methods, lubricants-types and applications, Lubrication methods, general sketch, working and applications, i. Screw down grease cup, ii. Pressure grease gun, iii. Splash lubrication, iv. Gravity lubrication, v. Wick feed lubrication vi. Side feed lubrication, vii. Ring lubrication, Definition, principle and factors affecting the corrosion. Types of corrosion, corrosion prevention methods.

Module-IV: (07 hrs)

Fault tracing: Fault tracing-concept and importance, decision tree concept, need and applications, sequence of fault-finding activities, show as decision tree, draw decision tree for problems in machine tools, hydraulic, pneumatic, automotive, thermal and electrical equipment's like, I. Any one machine tool, ii. Pump iii. Air compressor, iv. Internal combustion engine, v. Boiler, vi. Electrical motors, Types of faults in machine tools and their general causes.

Module-V: (08 hrs)

Periodic and preventive maintenance: Periodic inspection-concept and need, degreasing, cleaning and repairing schemes, overhauling of mechanical components, overhauling of electrical motor, common troubles and remedies of electric motor, repair complexities and its use, definition, need, steps and advantages of preventive maintenance. Steps/procedure for periodic and preventive maintenance of: I. Machine tools, ii. Pumps, iii. Air compressors, iv. Diesel generating (DG) sets, Program and schedule of preventive maintenance of mechanical and electrical equipment, advantages of preventive maintenance. Repair cycle concept and importance.

Books:

1. Maintenance Engineering Handbook, Higgins & Morrow, Da Information Services.

2. Maintenance Engineering, H. P. Garg, S. Chand and Company.
3. Pump-hydraulic Compressors, Audels, McGraw Hill Publication.
4. Foundation Engineering Handbook, Winterkorn, Hans, Chapman & Hall London.