VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE

(VJTI)

MATUNGA, MUMBAI 400 019

(Autonomous Institute affiliated to University of Mumbai)

CURRICULUM

(Scheme of Instruction & Evaluation and Course contents)

Second Year
of

Four Year Undergraduate Program

Bachelor of Technology (B Tech) Degree
in Computer Engineering

Implemented from the batch admitted in Academic Year 2024-25

B. Tech. Computer Engineering

Program Educational Objectives (PEOs)

PEO1. Achieve excellence in their profession and demonstrate leadership skills in multidisciplinary domain.

PEO2. Promote design, analysis, product implementation, research, and services in the field of Information Technology through strong technical, communication and entrepreneurial skills.

PEO3. To complement the class room teaching with live projects, fieldwork, seminars to build self-learning, and lifelong learning capability, and to develop out of box thinking.

Program Outcomes (POs)

PO1:	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2:	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3:	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4:	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5:	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6:	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7:	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8:	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9:	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10:	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11:	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a

	member and leader in a team, to manage projects and in multidisciplinary environments.
PO12:	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs) for B.Tech. In Computer Engineering

PSO1: Professional Skills: The ability to analyse, design and implement application specific computer engineering domains related to Big Data Systems, Cloud Computing, Artificial Intelligence, Machine Learning, Networking and Cyber Security applications for efficient design of computer based system of varying complexity by applying the knowledge of core science, engineering mathematics and engineering fundamentals.
PSO 2: Problem-Solving Skills: The ability to adapt and apply rapid changes in tools and technology in software development using open ended programming environment to deliver a quality product relevant to professional engineering practice through life-long learning.
PSO 3: Successful Career and Entrepreneurship: Excellent adaptability to function in multi-disciplinary work environment, good interpersonal skills as a leader in a team in appreciation of professional ethics, societal responsibilities and a zest for higher studies.

Department of Computer engineering and Information Technology

Credit Framework for UG Programme in Computer Engineering (Level 5.0- UG) - Semester - III

Sr.	Course Type	Course Code	Course Name	L	T	P	Hr	Cr	Examination Weightage in %			Ownership
									TA	MST	ESE	
1	PCC	R5MA2007T	Mathematics for Computer Engineers	3	0	0	3	3	20	30	50	Mathematics
2	PCC	R5CO2001T	Discrete Structure	3	1	0	4	4	20	30	50	CE & IT
3	PCC	R5CO2002T	Design & Analysis of Algorithm	3	0	0	3	3	20	30	50	CE & IT
4	PCC	R5CO2003T	Operating System	3	0	0	3	3	20	30	50	CE & IT
5	MDM		Multi-disciplinary Minor-I	2	0	0	2	2	20	30	50	Institute
6	PCC	R5CO2005L	Program Development Laboratory	0	0	2	2	1	ISCE :60		40	CE & IT
7	PCC	R5CO2002L	Algorithm Laboratory	0	0	2	2	1	ISCE :60		40	CE & IT
8	PCC	R5CO2003L	Operating System Lab	0	0	2	2	1	ISCE :60		40	CE & IT
9	AEC		Modern Indian Languages	2	0	0	2	2	ISCE :60		40	Humanities
10	VSEC	R5CO2004L	Open Source Technology Laboratory	0	0	2	2	1	ISCE :60		40	CE & IT
11	VEC	R5HS2401O	Universal Human Values	2	0	0	2	2	ISCE :60		40	
			Total	18	1	8	27	23				

Minor in Computer Engineering		**Subject Name**
1	Data Science	Introduction to Data Science
2	Cyber Security	Foundations of Cyber Security
3	Software Engineering	Introduction to Computer & Software
4	Advance Computing	Mathematical foundations of Computer Science

Open Electives for Computer Engineering
Automation and Robotics
Entrepreneurship Development

Credit Framework for UG Programme in Computer Engineering (Level 5.0- UG) - Semester - IV

Sr.	Course Type	Course Code	Course Name	L	T	P	Hr	Cr	Examination Weightage in %			Ownership
									TA	MST	ESE	
1	PCC	R5CO2006T	Theory of Computation	3	1	0	4	4	20	30	50	CE & IT
2	PCC	R5CO2007T	Artificial Intelligence	3	0	0	3	3	20	30	50	CE & IT
3	PCC	R5CO2008T	Database Management System	3	0	0	3	3	20	30	50	CE & IT
4	PCC	R5CO2009T	Software Engineering	3	0	0	3	3	20	30	50	CE & IT
5	MDM		Multi-disciplinary Minor-II	2	0	0	2	2	20	30	50	Institute
6	PCC	R5CO2007L	Artificial Intelligence Lab	0	0	2	2	1	ISCE :60		40	CE & IT
7	PCC	R5CO2008L	Database Management System Lab	0	0	2	2	1	ISCE :60		40	CE & IT
8	PCC	R5CO2009L	Software Engineering Lab	0	0	2	2	1	ISCE :60		40	CE & IT
9	VSEC	R5CO2010L	Linux administration Lab	0	0	2	2	1	ISCE :60		40	CE & IT
10	VEC	R5CH2402O	Environmental Science	2	0	0	2	2	ISCE :60		40	Chemistry
11	CEP/ FP	R5CO2601P	Community Engineering Project (Field Project)	0	0	4	4	2	ISCE :60		40	CE & IT
			Total	16	1	12	29	23				

Minor in Computer Engineering		**Subject Name**
1	Data Science	Python for Data Science and Data Analysis
2	Cyber Security	Cybersecurity Law & Privacy
3	Software Engineering	Foundations of Software Engineering
4	Advance Computing	Data Structure

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5MA2007T		
Course Title	Mathematics for Computer Engineers		
Course Scheme	**(L-T-P: 3-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	3		
Course Type	Program Core Course		
Prerequisites: Nil			

Course Outcomes:

At the end of the course student will be able to:

CO1. Solve systems of Linear equations using direct and iterative numerical methods and LU decomposition.

CO2. Apply Power method to find eigenvalues of a matrix.

CO3. Apply Gram- Schmidt process on Inner Product spaces to construct an orthonormal basis for a subspace and find QR factorization of a matrix.

CO4. Obtain the orthogonal and normal canonical reduction of a quadratic form.

CO5. Characterize complex variables in terms of analyticity, find harmonic conjugates and study geometric properties of conformal mappings.

CO6. Evaluate line integral of a complex function using parametrization and Cauchy's formula.

CO7. Identify and apply various properties of integers including the Well- Ordering Principle, unique factorization, Division Algorithm and solve the congruence equations.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Numerical Linear Algebra:** Solving systems of Linear equations using Gaussian elimination method, Iterative methods like Gauss seidel, Jacobi methods, LU decomposition, Crout's method, PLU decomposition, Power method to find eigenvalues.	1, 2	10
2	**Inner Product Spaces:** Properties, norm, Cauchy- Schwartz inequality, Orthogonal complements and projections, Orthonormal bases: Gram- Schmidt process, QR factorization, best approximation and Least squares.	3	5
3	**Quadratic forms:** Diagonalization of Symmetric matrices, Spectral theorem for symmetric matrices, Quadratic forms, Canonical forms, Positive definiteness and Cholesky's factorization of a matrix, application to Constrained optimization.	4	5
4	**Complex Analysis:** Complex numbers, Powers and roots of a complex number and their geometry, Complex Functions- Limits, Continuity and differentiability, Analytic functions, Hyperbolic functions, Cauchy Riemann equations, Harmonic conjugates, Conformal mappings, Linear fractional transformations and their properties, Line integrals, Cauchy's integral theorem and formula, derivatives of analytic functions.	5, 6	10

5	**Elementary Number Theory:** Divisibility and Modular arithmetic, Division algorithm, Primes, Fundamental Theorem of Arithmetic, Greatest Common Divisor, Euclidean Algorithm, Unique Factorization theorem, Solving congruences, Chinese remainder theorem, Well-ordering principle.	7	8

Text Books	
1	David Poole, Linear Algebra: A Modern Introduction, Third Edition.
2	S. Kumaresan, Linear Algebra: A Geometric Approach.
3	Discrete Mathematical Structures with Applications to Computer Science, J. P. Tremblay, R. Manohar, Tata McGraw Hill, 2008.
4	Complex variables and applications, J. W. Brown and R. V. Churchill, McGraw Hill.
5	Advanced Engineering Mathematics, Erwin Kreyszig, 9^{th} Edition, John Willey and sons.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2001T		
Course Title	Discrete Structure		
Course Scheme	**(L-T-P: 3-1-0) (TA-MST-ESE : 20-30-50)**		
Credits	4		
Course Type	Program Core Course		
Prerequisites: Nil			

Course Outcomes:

At the end of the course student will be able to:

CO1. Explain and interpret the basic ideas of propositional logic, set theory, and mathematical induction and application.

CO2. Use techniques of discrete probability, permutation, and combination to solve real-world problems in various domains.

CO3. Apply and analyze the characteristics of functions, equivalence relations, and binary relations, and their implications in computational contexts.

CO4. Construct and evaluate graphs and trees, and utilize graph algorithms to address issues related to networking, optimization, and data structures.

CO5. Evaluate and apply algebraic structures such as fields, rings, and groups in solving complex problems.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Foundations:** Sets Theory and Its Applications, Principle of inclusion exclusion, Propositional Logic, De Morgan's laws, Principle of mathematical induction, Predicates and quantifiers, Rules of Inference, Applications of Logic.	1	4
2	**Permutation, Combination and Discrete Probability:** The rules of Sum and Product, Permutation, Combination, Counting principles. Random experiment; sample space; events; axioms of probability; conditional probability. Theorem of total probability; Bayes' theorem. Application to information theory and discrete probability, Markov chains and their applications.	2	6
3	**Relations & Function:** Properties of Binary Relations, Closure of relations, Warshall's algorithm, Equivalence Relations and partitions, Partial ordering relations, Lattice, Functions and Pigeonhole Principle.	3	8
4	**Discrete Numeric Functions and Recurrence Relations:** Linear recurrence relations with constant coefficients (homogeneous and non-homogeneous); discussion of several special cases to obtain particular solutions. Solution of generating functions, Master Theorem for analyzing the time complexity of recursive algorithms.	3	6
5	**Graph & Tree:** Path, cycles, handshaking theorem, bipartite graphs, sub-graphs, graph isomorphism, operations on graphs, Eulerian graphs and Hamiltonian graphs, planar graphs, Euler formula, travelling salesman problem, Graph Coloring, shortest path algorithms (Dijkstra's algorithm).Basic terminology and characterization of trees, Prefix codes and optimal prefix codes, Binary search trees Tree traversal, Spanning trees and cut sets, Minimal Spanning trees, The Max flow-Min Cut Theorem (Transport network).	4	8

6	**Algebraic Structures:** Algebraic Systems, Groups, Semi Groups, Monoid, Subgroups, Permutation Groups, Codes and Group codes, Isomorphism and Automorphisms, Homomorphism, Homomorphism and Normal Subgroups, Ring, Integral Domain, Field, Ring Homomorphism, Fermat's Little Theorem, Polynomial Rings. Applications of Groups: error-correcting codes and cryptography.	4	6

Text Books	
1	Kenneth H Rosen, "Discrete mathematics & its Applications", Tata McGraw Hill Publication, 8th edition, 2018.
2	Liu C. L., "Elements of Discrete Mathematics", Six Edition, Tata McGraw-Hill, 2008, ISBN 10:0-07-066913-9.
Recommended Reading	
1	Jean-Paul Tremblay, R.Manohar, Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill.
2	Mott J. L, Kandel A. and Baker T. P., "Discrete Mathematics for Computer Scientists and Mathematicians", Second Edition, Prentice Hall India, 1986.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2002T		
Course Title	Design and Analysis of Algorithm		
Course Scheme	**(L-T-P: 3-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	3		
Course Type	Program Core Course		
Prerequisites: Data Structures			

Course Outcomes:

At the end of the course student will be able to:

CO1. Analyse, calculate and represent the time complexity and space complexity for given algorithm.

CO2. Understand the major graph algorithms and their analysis. Employ graphs to model engineering problems.

CO3. Identify and Analyse the problem statement and formulate solution using divide and conquer method.

CO4. Identify and Analyse the problem statement and formulate solution using Greedy technique.

CO5. Identify and Analyse the problem statement and formulate solution using Dynamic Programming technique.

CO6. Understand and Differentiate polynomial and NP problems and respective algorithmic aspects.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Introduction to Analysis of Algorithms**: Fundamentals of algorithmic problem solving, Fundamental of the analysis of algorithm: growth of functions, Time Complexity and Space Complexity, asymptotic notation, Recurrences, Solving Recurrences and time equations: substitution method, the recursion-tree method and the master method. Brief introduction to different algorithm designs. **Sorting Algorithms and their analysis**: Selection Sort, Insertion Sort, and Heap Sort.	1	6
2	**Graph Algorithms** : Minimum spanning trees algorithms: Kruskal's Minimum Spanning Tree Algorithm, Prim's Minimum Spanning Tree Algorithm , Dijkstra's Single Source Shortest path algorithms, all pair shortest path algorithm, Applications of graph algorithms, Use of Graph Algorithms for topological sort, connected components etc.	2	5
3	**Divide and Conquer Technique**: General Method, Merge sort, Quick sort, Binary search and their algorithmic analysis, counting inversions, finding closest pair of points, Karatsuba algorithm for Integer multiplication, Finding the maximum-subarray problem.	3	6
4	**Greedy Method**: Elements of greedy technique, Activity selection problem, Fractional Knapsack Problem, Job Sequencing problem, Huffman Coding, Finding Understanding and Analysing the graph algorithms as the greedy algorithms: Dijkstra Algorithm, Minimum Spanning Tree Kruskal's and Prim's algorithm.	2, 4	6

5	**Dynamic Programming** Elements of Dynamic Programming, The Principle of Optimality, Assembly Line-Scheduling, and Matrix chain multiplication, Longest Common Subsequence, 0-1 Knapsack problem, Understanding and Analysing the All Pair Shortest Path with dynamic programming technique, Making Change with coins Problem.	2, 5	6
6	**NP-completeness** The class P and NP, NP-completeness, Polynomial-time verification, NP-completeness and reducibility, NP-complete problems such as Search/Decision, SAT, 3VC, Clique, Vertex Cover, Subset Sum. Approximation Algorithms: vertex-cover problem, Subset-sum Problem, Set-covering problem.	6	6

Text Books	
1	T. H. Cormen, C. E. Leiserson, R. L.Rivest and C. Stein, "Introduction to Algorithms", MIT Press/McGraw Hill, Second Edition / Forth Edition.
2	Jon Kleinberg, Eva Tardos, "Algorithm Design", Pearson, Addison Wesley.
3	Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms", University Press.
Recommended Reading	
1	A. V. Aho, J. E Hopecroft and J.D. Ullman, The design and analysis of algorithm, Addision-Wesley, 1974.
2	Design and Analysis of Algorithms, Dave and Dave, Pearson.
3	Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2003T		
Course Title	Operating System		
Course Scheme	**(L-T-P: 3-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	3		
Course Type	Program Core Course		
Prerequisites: Python programming, Computer Organization			
Course Outcomes: At the end of the course student will be able to: **CO1.** Understand Operating systems basics. **CO2.** Analyse various process management concepts including scheduling, synchronization, and deadlocks. **CO3.** Understand multithreading, and analyse the issues related to file system. **CO4.** Apply memory management and system resource-sharing concepts. **CO5.** Examine I/M Management. **CO6.** Evaluate Operating System Security.			

Course Contents		CO Mapped	Teaching Hours/Week
1	**Introduction:** Computers and Software, General System software, Resource abstraction & Sharing, Operating system strategies (Batch, Timesharing, real-time, embedded, etc.)Concept of Multiprogramming Operating system organization, Basic functions, Implementation considerations, Computer organization, bootstrapping the machine, Mobile computers, Multiprocessors and parallel computers, Device Management-Device controllers & Device drivers, I/O strategies (direct I/O with polling, Interrupt driven I/O, DMA), Buffering, Disk scheduling strategies.	1	5
2	**Process and Threads Management:** Process & Threads-Implementing process & Threads – Process address space- process state transition diagram- Process manager responsibilities- the concept of Linux process & thread descriptors-Process scheduler organization- different scheduling strategies(Non-preemptive & preemptive)- Process synchronization- critical section- semaphore & its implementation – classical synchronization problems and its solutions (Producer consumer, readers-writers, dining philosopher)- Deadlock-prevention-avoidance-bankers algorithm-detection-reduced resource allocation graph- Inter-process communication(Pipes, message passing etc.) - concept of process management in Linux and Windows NT.	2	6
3	**Memory Management:** Memory management- address space abstraction-address binding-memory allocation- Fixed partition & variable partition memory strategies-dynamic address binding swapping- paging-virtual memory address translation-dynamic paging static paging algorithms-dynamic paging algorithm-working set algorithm-segmentation- implementation-memory mapped files-concept of memory management in Linux & Windows NT/XP.	3	5
4	**File Management:** File Management – Low-level files and Structured files-Low-level file implementation .– different approaches to Block management-Structured sequential file-Indexed sequential file, different directory structures,	4	5

	file systems, Mounting file systems- Protection and Security, security and Policy , Authentication , authorization and cryptography- Kerberos authentication- General protection model- Access matrix- Access control list – Capability list – Concept of File management in Linux and Windows NT.		
5	**I/O Management:** I/O Devices, Organization of the I/O Function, Operating System Design Issues, Buffer Management.	5	2
6	**Architecture of the UNIX:** Introduction to system concepts, Kernel data structures, system administration.	6	2
7	**Buffer Cache:** Structure of the buffer pool, Advantages and disadvantages of the buffer cache. Internal representation of files: Inodes. Structure regular file. Directories – Conversion of a path name to an I-node. Super block. Other file types, Unified Buffer Cache.	6	3
8	**Protection and Security:** Goals of protection, Domain of protection, Access matrix, Implementation of access matrix, Revocation of access rights, Security problems, Authentication, Program threats, System threats, Threat monitoring.	6	3
9	**Overview of Mobile Operating** Systems Definition and characteristics History and evolution Types of Mobile Operating Systems Android OS Windows Phone Others (e.g., BlackBerry, Symbian) Overview of Real-Time Operating Systems (RTOS) Definition and characteristics Types of real-time systems (hard vs. soft)	6	3

Text Books	
1	Silberschatz & Galvin, “Operating system concepts”, Addison Wesley ,10th edition.
2	Tanenbaum A.S, “Modern Operating Systems”, Pearson Education 3rd edition, 2008.
3	William Stallings, Operating Systems: Internals and Design Principles, Prentice Hall,2008.
Recommended Reading	
1	Gary Nutt, Nebendu Chaki, and Sarmistha Neogy, “Operating Systems”,Pearson Education, 3rd edition, 2009.
2	Jerry D. Peek, Grace Todino, John Strang, “Learning the Unix Operating System”,O'Reilly & Associates Publication, 5th edition, 2002.
3	Crowley C., “Operating Systems – A Design oriented Approach”, TMH.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2201T		
Course Title	Multi-disciplinary Minor-I: Introduction to Data Science		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-I (Data Science)		
Prerequisites: Nil			
Course Outcomes: At the end of the course student will be able to: **CO1.** To Provide the knowledge and expertise to become a proficient data scientist. **CO2.** Demonstrate an understanding of statistics and machine learning concepts that are vital for data science. **CO3**. Produce Python code to statistically analyse a dataset.			

Course Contents		CO Mapped	Teaching Hours/Week
1	**Introduction to Data Science** Different Sectors using Data science, Purpose and Components of Python in Data Science.	1	4
2	**Data Analytics Process** Knowledge Check, Exploratory Data Analysis (EDA), EDA- Quantitative technique, EDA- Graphical Technique, Data Analytics Conclusion and Predictions.	2	6
3	**Feature Generation and Feature Selection** Extracting Meaning from Data-Motivating application: user (customer) retention- Feature Generation (brainstorming, role of domain expertise, and place for imagination)- Feature Selection algorithms.	3	6
4	**Data Visualization** Basic principles, ideas and tools for data visualization, Examples of inspiring (industry) projects- Exercise: create your own visualization of a complex dataset.	3	8

Text Books	
1	Business Analytics: The Science of Data - Driven Decision Making, U Dinesh Kumar, John Wiley & Sons.
2	Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, John Wiley & Sons.
Recommended Reading	
1	Cathy O'Neil and Rachel Schutt. Doing Data Science, Straight Talk from The Frontline. O'Reilly Publisher.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2202T		
Course Title	Multi-disciplinary Minor-I Foundations of Cyber Security		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-I (Cyber Security)		
Prerequisites: Nil			

Course Outcomes:

At the end of the course student will be able to:

CO1. To demonstrate an understanding of foundational cybersecurity principles and concepts.

CO2. To implement security measures to safeguard information assets, ensure data integrity and availability, authenticate users securely.

CO3. To understand the legal and ethical implications of cybersecurity practices and policies.

CO4. To apply basic security measures to protect digital assets and mitigate cybersecurity risks.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Introduction to Cyber Security** Overview of cybersecurity concepts, Introduction, Computer Security, Threats, Harm, Vulnerabilities, Controls, Authentication, Web attacks: Browser Attacks, Web Attacks Targeting Users, Obtaining User or Website Data, Email Attacks. Importance of cybersecurity in modern society, Historical perspective and evolution of cybersecurity. Cyber-attack vectors and methodologies, Impact of cyber threats on individuals, organizations, and society.	1	6
2	**Cyber Security Fundamentals** Confidentiality, integrity, and availability (CIA) triad, Authentication and authorization, Non-repudiation and accountability, Encryption algorithms and protocols, Access control mechanisms: RBAC, MAC, DAC.	1,2	6
3	**Introduction to Cyber Crime, law and Investigation** Cyber Crimes, Types of Cybercrime, Hacking, Attack vectors, Cyberspace and Criminal Behaviour, Clarification of Terms, Traditional Problems Associated with Computer Crime, Introduction to Incident Response, Digital Forensics, Computer Language, Network Language, Realms of the Cyber world. Internet crime and Act: A Brief History of the Internet, Recognizing and Defining Computer Crime, Contemporary Crimes, Computers as Targets, Contaminants and Destruction of Data, Indian IT ACT, 2000. Firewalls and Packet Filters, password Cracking, Key loggers and Spyware, Virus and Warms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQL injection, Buffer Overflow, Attack on wireless Networks.	3	6
4	**Cyber Security Practices and Tools** Password management best practices, Multi-factor authentication (MFA) and its implementation, Secure coding principles, Common vulnerabilities in code (e.g., buffer overflows, injection attacks), Mitigating security risks in software development, Intrusion detection and prevention systems (IDPS), Introduction to Firewalls and Network Security, Secure Communication Protocols.	4	6

Text Books	
1	Cybersecurity Essentials" by Charles J. Brooks (2nd Edition).
2	"Cybersecurity – Attack and Defense Strategies" by Yuri Diogenes and Erdal Ozkaya (1st Edition).
Recommended Reading	
1	"Foundations of Cybersecurity" by Steve Winterfeld and Jason Andress (1st Edition).
2	"Principles of Cybersecurity" by Steven Hernandez (1st Edition).

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2203T		
Course Title	Multi-disciplinary Minor-I **Introduction to Computer & Software**		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-I (Software Engineering)		
Prerequisites: Nil			

Course Outcomes:
At the end of the course student will be able:

CO1. Explain the basics of Computer, its components, peripherals and different generations of Computers.
CO2. Identify different types of computer memory and Input/Output Devices.
CO3. Demonstrate the knowledge of Computer Programming languages & Computer software.
CO4. Classify different Systems & Application Software.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Overview of Computer and its Components:** History and generations of computers, types of computers, technological advancements, and their impact on computing. Introduction to computer hardware, central processing unit (CPU), Primary Memory, Secondary storage devices, Input/output system and peripherals, working of computer.	1	4
2	**Primary Memory, Secondary Storage, Input/output Devices:** Types of memory (RAM, ROM), storage devices (HDD, SSD), and various input/output devices.	2	4
3	**Computer Programing, Programming Languages & Computer Hardware & Software:** Introduction to programming concepts, different programming languages, and introduction to computer software.	3	4
4	**Systems and Application Software:** Study of systems software, operating systems, Device Drivers, Firmware, Utility Software, Boot loaders, program loader, linker, debugger, compiler, interpreter, utility programs and application software.	24	4

Text Books	
1	Anita Goel. Computer Fundamentals, Pearson Education, first edition, 2010.
2	P. K. Sinha & P. Sinha. Fundamentals of Computers, BPB Publishers 2007.
Reference Books	
1	B. Ram, Sanjay Kumar. Computer Fundamentals: Architecture and Organization, New Age International Publishers, fifth edition 2018

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2204T		
Course Title	Multi-disciplinary Minor-I Mathematical foundations of Computer Science		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-I (Advance Computing)		
Prerequisites: Nil			
Course Outcomes: At the end of the course student will be able: **CO1.** To understand the basic notions of discrete and continuous probability. **CO2.** To understand the methods of statistical inference, and the role that sampling distributions play in those methods.			

Course Contents		CO Mapped	Teaching Hours/Week
1	**Probability mass** density, and cumulative distribution functions, Parametric families of distributions, Expected value, variance, conditional expectation, Applications of the univariate and multivariate Central Limit Theorem.	1	6
2	**Random samples** sampling distributions of estimators, Methods of Moments and Maximum Likelihood.	1	6
3	**Statistical inference** Introduction to multivariate statistical models: regression and classification problems, principal components analysis, The problem of overfitting model assessment.	2	6
4	**Graph Theory** Isomorphism, Planar graphs, graph colouring, Hamilton circuits and Euler cycles. Permutations and Combinations with and without repetition. Specialized techniques to solve combinatorial enumeration problems	2	6

Text Books	
1	John Vince, Foundation Mathematics for Computer Science, Springer.
Recommended Reading	
1	K. Trivedi. Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Wiley.
2	M. Mitzenmacher and E. Upfal.Probability and Computing: Randomized Algorithms and Probabilistic Analysis.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2005L		
Course Title	Program Development Laboratory		
Course Scheme	**(L-T-P: 1-0-2) (ISCE:60, 40)**		
Credits	2		
Course Type	Program Core Course		
Prerequisites: Programming Fundamentals			

Course Outcomes:
At the end of the course student will be able to:
CO1. Understand and apply fundamental programming concepts using Python.
CO2. Implement and utilize various data structures and algorithms in Python for problem-solving.
CO3. Design and develop Python programs using modular and object-oriented programming principles.
CO4. Analyse and debug Python code effectively to create robust software solutions.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Introduction to Python Programming** Overview of Python and its features Installing Python and setting up the development environment, Basic syntax, data types, and variables, Input/output operations in Python.	1	3
2	**Control Flow and Functions** Conditional statements (if, elif, else), Loops (for and while loops), Functions and parameter passing, Recursion and its applications.	1	4
3	**Data Structures in Python** Lists, tuples, sets, and dictionaries, Operations and methods on data structures, List comprehensions and generator expressions, Working with nested data structures.	2	4
4	**File Handling and Modules** Reading from and writing to files in Python, File handling modes and operations, Creating and using Python modules, Importing modules and packages in Python.	2	3
5	**Object-Oriented Programming (OOP)** in Python Introduction to OOP concepts: classes and objects, Encapsulation, inheritance, and polymorphism Class constructors, methods, and attributes, Overriding methods and operator overloading.	3	4
6	**Introduction** to popular Python libraries NumPy, pandas.	4	2

Text Books	
1	Introduction to Python for Computational Science and Engineering (A beginner's guide), Hans Fangohr.
2	Exploring Python, Timothy A. Budd, McGraw Hill Education.
Recommended Reading	
1	Python for Informatics: Exploring Information, Charles Severance.
2	Learning Python, Fourth Edition, Mark Lutz, O'Reilly publication.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2002L		
Course Title	Design and Analysis of Algorithm Lab		
Course Scheme	**(L-T-P: 0-0-2) (ISCE:60, 40)**		
Credits	1		
Course Type	Program Core Course		
Prerequisites: Data Structures			

Course Outcomes:
At the end of the course student will be able to:
CO1. Implement Sorting and Searching Algorithms and Analyse the time complexity and space complexity for given algorithm.
CO2. Understand the major graph algorithms and their analysis. Employ graphs to model engineering problems.
CO3. Design Algorithms using Divide and Conquer methods for given problem statement.
CO4. Design Algorithms using Greedy methods for given problem statement and analyse their efficiency.
CO5. Design Algorithms using Dynamic Programming for given problem statement and analyse their efficiency.
CO6. Understand and Implement Algorithms using Heuristics, Approximation methods for given NP-problem statement and analyse their efficiency.

Course Contents		CO Mapped	Teaching Hours/Week
1	A. Implement various searching algorithms and exercises using searching algorithms. B. Implement various sorting algorithms. C. Comparison of sorting algorithms based on time complexity and demonstrate it using graphical representation.	1	4
2	A. Design and Implement Graph related exercises using traversing algorithms. B. Design and Implement Graph related exercises using Minimum spanning tree algorithms.	2	4
3	A. Design and Implement solutions for problem statements using divide and conquer method. B. Implement solutions for at least 2 problems and analyse the improvement in time complexity and space complexity.	3	4
4	A. Design and Implement solutions for problem statements using greedy method. B. Implement solutions for at least 2 problems and analyse the improvement in time complexity and space complexity.	2, 4	4
5	A. Design and Implement solutions for problem statements using dynamic programming. B. Implement solutions for at least 2 problems and analyse the time complexity and space complexity.	2, 5	4

6	Understand NP-completeness of the problem and implement the approximation algorithm as the solution of the problem.	6	4

Text Books	
1	T. H. Cormen, C. E. Leiserson, R. L.Rivest and C. Stein, "Introduction to Algorithms", MIT Press/McGraw Hill, Second Edition / Forth Edition.
2	Jon Kleinberg, Eva Tardos, "Algorithm Design", Pearson, Addison Wesley.
3	Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms", University Press.
Recommended Reading	
1	A. V. Aho, J. E Hopecroft and J.D. Ullman, The design and analysis of algorithm, Addision-Wesley, 1974.
2	Design and Analysis of Algorithms, Dave and Dave, Pearson.
3	Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2003L		
Course Title	Operating System Laboratory		
Course Scheme	**(L-T-P: 0-0-2) (ISCE:60, 40)**		
Credits	1		
Course Type	Program Core Course		
Prerequisites: Nil			

Course Outcomes:
At the end of the course student will be able to:
CO1. To execute OS commands, shell scripts.
CO2. To compare the performance of various CPU Scheduling Algorithm, file allocation algorithms.
CO3. To analyse the performance of the various page replacement algorithms, deadlock avoidance and detection.
CO4. To create processes and implement IPC.

List of Experiments		CO Mapped	Teaching Hours/Week
1	Basics of UNIX commands.	1	1
2	Shell Programming.	1	1
3	Implement the following CPU scheduling algorithms Round Robin SJF FCFS Priority.	2	1
4	Implement all file allocation strategies Sequential Indexed Linked.	2	1
5	Implement Semaphores.	3	1
6	Implement all File Organization Techniques.	2	1
7	Implement Bankers Algorithm for Dead Lock Avoidance.	3	1
8	Implement an Algorithm for Dead Lock Detection.	3	1
9	Implement e all page replacement algorithms FIFO, LRU, LFU.	3	1
10	Implement Shared memory and IPC.	4	1
11	Implement Paging Technique of memory management.	3	1
12	Implement Threading & Synchronization Applications.	4	1
13	Write a C program to simulate producer-consumer problem using semaphores.	3	1

Text Books	
1	Silberschatz & Galvin, “Operating system concepts”, Addison Wesley ,10th edition.
2	Tanenbaum A.S, “Modern Operating Systems”, Pearson Education 3rd edition, 2008.
3	William Stallings, Operating Systems: Internals and Design Principles, Prentice Hall, 2008.
Recommended Reading	
1	Gary Nutt, Nebendu Chaki, and Sarmistha Neogy, “Operating Systems”, Pearson Education, 3rd edition, 2009.
2	Jerry D. Peek, Grace Todino, John Strang, “Learning the Unix Operating System”, O’Reilly & Associates Publication, 5th edition, 2002.
3	Crowley C., “Operating Systems – A Design oriented Approach”, TMH.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5CO2004L		
Course Title	Open source Tools & Technology Lab		
Course Scheme	**(L-T-P: 0-0-2) (ISCE:60, 40)**		
Credits	1		
Course Type	Vocational and Skill Enhancement Course		
Prerequisites: Data Structures, Programming Language			
Course Outcomes: At the end of the course student will be able to: **CO1.** Install, configure, and use a variety of open-source tools. **CO2.** Utilize open-source IDEs and text editors for software development. **CO3.** Collaborate on open-source projects using platforms like GitHub and GitLab. **CO4.** Develop and manage projects using version control systems.			

Course Contents		CO Mapped	Teaching Hours/Week
1	**Open Source Technologies** Introduction to open source technology, Internet, open source operating systems, open source platform.	1	4
2	**Open source development** Demographics, sociology and Psychology of open source development, legal issues in open source, Economics of open source, the GNU Project.	2	4
3	**Version Control Systems** Introduction to Git, Basic Git commands (clone, commit, push, pull, branch, merge), Working with GitHub/ GitLab.	2, 3, 4	4
4	**Development Environments** Introduction to Integrated Development Environments (IDEs) and Text Editors, Using VSCode, Atom, and Sublime Text, Customizing development environments.	2, 3, 4	4
5	**System Monitoring and Logging** Monitoring systems with Nagios/Zabbix, Logging with ELK stack (Elastic search, Log stash, Kibana), Performance analysis and tuning.	2, 3	4
6	**Collaboration and Contribution** Open-source project workflows, Contributing to open-source projects, Case studies of successful open-source projects.	1, 3	4

Text Books	
1	Scott Chacon, Ben Straub , Pro Git, by Apress Open 2014.
2	Open Source: Technology and Policy, Fadee P. deek, James McHugh, 2008.
3	Official documentation of tools covered in the course (e.g., Git, Docker, Jupyter).
Recommended Reading	
1	MEAN frame work, web source.

Programme Name	B. Tech. Computer Engineering	**Semester**	III
Course Code	R5HS2401O		
Course Title	Universal Human Values		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	1		
Course Type	Value Education Course		
Prerequisites:			

Course Outcomes:

At the end of the course student will be able to:

CO1. Analyze the significance of value inputs provided in formal education along with skills and develop a broader perspective about life and education.

CO2. Formulate their aspirations and concerns at different levels of living, and the way to 25odell them them in a sustainable manner.

CO3. Evaluate their current state of understanding and living, and model a healthy lifestyle.

CO4. Examine the issues of home sickness, interactions with seniors on the campus, peer pressure with better understanding and feel grateful towards parents, teachers and others.

CO5. Develop more confidence and commitment for value-based living in family, society and nature.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Aspirations and concerns** – Understanding basic human aspirations, fixing one's goals, and the need for a holistic perspective in form of Universal Human values, Self management – self confidence, handling peer pressure, time management, anger, stress, personality development and self improvement which leads to harmony in the human being.	1	5
2	**Understanding Health** – Health issues, healthy diet, healthy lifestyle which shall lead to Harmony of the self and body in forms of mental and physical health.	2	5
3	**Relationships** – Learning to handle home sickness, gratitude towards parents, teachers and others, understanding impact of ragging and interaction, competition and cooperation to achieve harmony in relationships.	3	6
4	**Participation in society**, participation in nature leading to harmony in the society and nature/existence, Role of education in developing holistic perspective	4,5	6

Text Books	
1	E.P.G.Gohl, L.D.Vilensky, Textile Science, an Explanation of Fibre Properties, Second Edition, 1987, CBS Publishers & Distributors Pvt. Ltd.
2	Manufactured Fibre Technology, V.B. Gupta and V.K. Kothari, Springer Science + Business Media, 2003, ISBN 978-94-010-6473-6
Recommended Reading	
1	Tatsuya Hongu, Glyn O. Phillips, MachikoTakigam, New Millennium Fibers, Woodhead Publishing Ltd., CRC Press LLC, 2005, ISBN 0-8493-2598-6.

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2006T		
Course Title	Theory of Computation		
Course Scheme	**(L-T-P: 3-0-1) (TA-MST-ESE : 20-30-50)**		
Credits	4		
Course Type	Program Core Course		
Prerequisites: Data Structures			

Course Outcomes:

At the end of the course student will be able to:

CO1. Explain and differentiate between various types of automata, including finite automata, pushdown automata, and Turing machines, as well as the formal languages associated with each.

CO2. Construct regular expressions and languages, and apply analytical techniques to evaluate their properties and limitations.

CO3. Design context-free grammars and parse trees, and apply minimization techniques to simplify the analysis of context-free languages.

CO4. Implement pushdown automata and establish the equivalence between finite automata and regular grammars, as well as between pushdown automata and context-free grammars.

CO5. Understand the concepts of computability, decidability, and un-decidability, and analyse the complexity of problems using the concepts of P, NP, NP-complete, and NP-hard problems.s

Course Contents		CO Mapped	Teaching Hours/Week
1	**Fundamentals & Finite Automata:** Strings, Alphabet, Language, Operations, Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, Deterministic finite automaton and Non-deterministic finite automaton, Transition diagrams, Language recognizers, Transforming NFA to DFA, NFA with epsilon moves, Minimization of DFA, Finite state automata with output – Moore and Mealy machine.	1	6
2	**Language & Grammar:** Regular sets, Closure properties of regular sets, regular expressions, identity rules, constructing finite Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, Formal Grammar, Types of Grammar, Chomsky Hierarchy, Regular Grammar, Equivalence of Regular Grammars and Finite Automata, Left Linear and Right Linear Grammar.	2	6
3	**Context Free Grammar:** CFG, Derivation, Parse Tree, Ambiguity in grammars and languages, Language Specification using CFG, Minimization of Context Free Grammars. Chomsky normal form, Greiback normal form, Pumping Lemma for Context Free Languages. Closure properties of CFL	3	8
4	**Pushdown automata:** Pushdown automata Definition, Model, Equivalence of Pushdown automata and CFG, Deterministic Pushdown Automata, Non-Deterministic Pushdown Automata, Graphical notation for PDA, two stack PDA.	4	6
5	**Turing Machine:** Definition, Model, Design of Turing Machine, Computable Functions, Programming Techniques for Turing Machine, Variation of Turing Machine. Turing Machine as an Integer Function, Linear-Bounded Automata (LBA)	4,5	6

6	**Computability and Un-decidability:** Decidability of problems, Universal Turing Machine, TM Language, Unrestricted Grammar, Modified Chomsky Hierarchy, Properties of Recursive and Recursively Enumerable Languages, Un-decidability, Post's Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.	5	8

Text Books	
1	J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", Pearson Education, second Edition, 2007.
2	Shyamalendu Kandar, "Introduction to Automata Theory, Formal Languages and Computation", Pearson Education, second Edition, 2007.
Recommended Reading	
1	Micheal Sipser, "Introduction of the Theory and Computation", Thomson Brokecole, 1997.
2	John Martin, "Introduction to Languages and the Theory of Computation", McGraw-Hill, 2003.

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2007T		
Course Title	Artificial Intelligence		
Course Scheme	**(L-T-P: 3-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	3		
Course Type	Program Core Course		
Prerequisites: Design and Analysis of Algorithms			

Course Outcomes:

At the end of the course student will be able to:

CO1. Understand artificial intelligence techniques, including search heuristics, knowledge representation, planning and reasoning.

CO2. To solve problems by applying a suitable search method, and AI applications in Natural Language Processing, Vision and Robotics.

CO3. Compare mini-max search and alpha-beta pruning in game playing.

CO4. Implement pushdown automata and establish the equivalence between finite automata and regular grammars, as well as between pushdown automata and context-free grammars.

CO5. Differentiate the key aspects of evolutionary computation, including genetic algorithms and genetic programming.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Introduction:** Overview and historical perspective, Turing test, physical symbol systems and the scope of symbolic AI, Agents.	1	3
2	**Searching techniques:** State Space Search: Depth First Search, Breadth first Search, DFID. Heuristic Search Best First Search, Hill Climbing, Beam Search, Taboo Search. Randomized Search Simulated annealing, Genetic Algorithms, Ant colony optimization.	1,2	4
3	**Finding Optimal Paths:** Branch and Bound, A*, IDA*, Divide and Conquer approaches, Beam Stack Search.	3	4
4	**Problem Decomposition:** Goal Trees, AO*, Rule Based Systems, Rete Net.	3,4	5
5	**Game Playing:** Minimax Algorithm, Alpha Beta Algorithm, SSS*.	4	5
6	**Planning and Constraint Satisfaction:** Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Constraint Propagation.	3	5
7	**Logic and Inferences:** Propositional Logic, First Order Logic, Soundness and Completeness, Forward and Backward chaining.	4	2
8	**AI Applications:** AI applications in Natural Language Processing, Vision and Robotics**.**	4	2
9	Advances in AI	4	2

Text Books

1	Deepak Khemani,"A First Course in Artificial Intelligence", McGraw Hill Education (India), 2013.
2	Stuart Russell, Peter Norvig, "Artificial Intelligence A Modern Approach", Prentice Hall, 3rd Edition, 2009.
Recommended Reading	
1	Stefan Edel Kamp and Stefan Schroedl, "Heuristic Search: Theory and Application", Morgan Kaufmann, 2011.
2	Zbigniew Michalewicz and David B. Fogel, "How to Solve it: Modern Heuristics", Springer, 2nd Edition, revised and extended edition, 2013.

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2008T		
Course Title	Database Management System		
Course Scheme	**(L-T-P: 3-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	3		
Course Type	Program Core Course		
Prerequisites: Data Structures			

Course Outcomes:

At the end of the course student will be able to:

CO1. Understand foundational concepts in database management systems, including data models, database system environment, client-server architectures.

CO2. Acquire proficiency in SQL (Structured Query Language), enabling them to write complex queries for data retrieval, manipulation, and management in relational database systems.

CO3. Design database using ER model.

CO4. Identify the database problems and perform normalization technique.

CO5. Identify issues in data storage, transaction, and concurrency control of DBMS, recover and security of databases.

CO6. Optimize database performance, including indexing strategies, query optimization, database tuning, and managing database resources effectively to enhance system scalability and responsiveness.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Introduction to DBMS**: Characteristics of database , Database users , Advantages of DBMS , Data Models (Relational, Non-Relational) , Schemas and Instances , three schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client / Server Architecture for DBMS.	1	6
2	**Structured Query Language:** DDL Create, Modify, Alter, Drop, View definition, etc. DML : SELECT, INSERT, DELETE, Update, Nested Query, SQL with SET operations: Union, Intersect, Except, etc, Aggregate Functions: Group By, Having, SUM, etc, SQL with Logical operations, Nested and Complex Queries, Join Queries. DCL: GRANT, REVOKE, etc DBA level query. Cursors and Triggers, Procedures and Functions.	2	7
3	**Entity-Relationship Model:** Entity Types ,Entity Sets ,Attributes and Keys, Relationship Types, Relationship sets , Roles and structural Constraints; Design Issues; Entity Relationship diagram; Weak entity sets; Extended E- R features; Design of an E-R database schema; Reduction of an E-R schema to tables. Relational Model Concept of a relation; Relational Model Constraints; Relational Database Schema, Entity Integrity, Referential Integrity and foreign keys; the relational algebra and extended relational- algebra operations; Relational Database Design using ER-to Relational Mapping.	3	5
4	**Relational-Database Design:** First normal form; Pitfalls in relational- database design ;Functional dependencies; Decomposition; Desirable properties of decomposition; Boyce-Codd normal form; 3rd and 4th normal form; Mention of	4	6

	other normal forms; Overall database design process.		
5	**Transaction Processing Concepts and Concurrency Control Protocols:** Transactions: Transaction concept; Transaction and System Concepts Properties of Transaction; Schedules based on Serializability; Recoverability; Transaction definition in SQL. Concurrency Control: Concurrency Control Lock-based protocols; Timestamp-based protocols; Validation-based protocols; Multiple granularities; Multiversion schemes; Deadlock handling; Insert and delete operations; Weak levels of consistency; Concurrency in index structures, locking techniques.	5	5
6	**Database Recovery :**Database back up, Recovery System Failure classification; Storage structure; Recovery and atomicity; Log-based recovery; Shadow paging; Recovery with concurrent transactions; Buffer management, database security issues, access control, authorization, Distributed databases.	5	5
7	**Data Storage and Querying:** File organization, Indexing and Hashing Organization of records in files; Data dictionary storage. Basic Concepts of Indexing ; Types of Single Level Ordered Indices; Multilevel Indices using B+ Tree Index Files; B- Tree Index Files; Static Hashing; Dynamic Hashing; Index Definition in SQL; Multiple-Key Access. Fundamentals of Query Optimization.	6	5
8	**Database security :** Introduction to Database Securities, Common threats and challenges, Best Practices, Tools and Platforms for securing data, SQL Injection Attacks and Prevention, DAC, MAC, RBAC.	5	4

Text Books	
1	Elmasri & Navathe , "Fundamentals of Database System", 7thEdition, Addison Wesely Publication.(2015).
2	Abraham Silberschatz, Henry Korth, Sudarshan , "Database System Concepts", 6th Edition, (2010).
3	Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 3rdEdition, McGraw-Hill, 2002.
Recommended Reading	
1	Michael Mannino, "Database design, Application Development and Administration", ,4th Edition(2008).
2	Peter Rob and Coronel, "Database systems, Design, Implementation and Management", 5th Edition, Thomson Learning, 2001.
3	C. J. Date, "Introduction To Database Systems", Seventh Edition, Addison Wesley Longman.

Programme Name	B.Tech in Computer Engineering	**Semester**	IV
Course Code	R5CO2009T		
Course Title	Software Engineering		
Course Scheme	**(L-T-P: 3-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	3		
Prerequisites	Computer & Software Fundamentals		

COURSE OUTCOMES:
At the end of this course, students will be able to
CO1. Demonstrate basic knowledge in Software Engineering. **CO2.** Estimate cost, effort & time, Manage risk, Prepare project schedule and track it for a software development project. **CO3.**Assure the quality of a software & Manage software configuration **CO4.**Plan, gather requirements for, 32odelli, design, develop and test the software development project.

Course Contents		HRS	CO
1	**Introduction** The Product: Software Characteristics, Applications. The Process: Software Process, Software Process Models, Linear Sequential model, Prototyping model, RAD model, Evolutionary models – Incremental model, Spiral model.	5	1
2	**Software project management concepts** Important factors of project management- People, Product, Process, Project.	2	1
3	**Software Process and Project Metrics** Measures, Metrics, Indicators. Metrics in the process and project domains, Software measurement. Metrics for Software Quality. Integrating metrics.	3	2
4	**Software Project Planning** Software Scope, Resources, Software project estimation – cost/effort estimation, Decomposition techniques, Empirical estimation models.	3	2
5	**Risk Analysis and Management** Reactive versus proactive risk strategies, Software risks, Risk identification, Risk projection, Risk mitigation-monitoring-management, RMMM plan.	5	2
6	**Project Scheduling and Tracking** Defining a task set for the software project, Gantt Chart, Defining a task network, Scheduling.	3	2

7	**Software Quality Assurance** Software quality assurance, Software reviews, Formal technical reviews, SQA plan.	3	3
8	**Software Configuration Management** SCM process, Identification of objects in the software configuration, Version control, Change control, Configuration audit, Status reporting.	3	3
9	**Analysis Concepts and Principles** Requirement Analysis, Requirement elicitation for software, Analysis principles, Software prototyping, Requirements Specification.	3	4
10	**Analysis Modeling** Data 33odelling, Functional 33odelling and information flow, Behavioral 33odelling.	2	4
11	**Design Concepts and Principles** Software design process, Design principles, Design concepts, Effective modular design.	4	4
12	**Design Modeling** Data Design. Architectural Design: Software architecture, Mapping requirements into a software architecture. User Interface Design: Human Factor, User interface design process. Component-Level Design: Structured programming design notations.	3	4
13	**Software Testing Techniques & Strategies** White-box & Black-box testing techniques. Strategic Approach to Software Testing.	3	4

Text Books:	
1.	Roger Pressman, "Software Engineering", McGraw Hill, Eighth Edition.
2	Ian Somerville, "Software Engineering", Pearson Education. Sixth Edition
Reference Books:	
1	James Peter ,"Software Engineering an Engineering approach", John Wiley, First Edition
2	W. S. Jawadekar, "Software Engineering", TMH. 1st Edition
3	R. Mall, "Fundamentals of Software Engineering", Prentice Hall of India, 2nd Edition

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2205T		
Course Title	Multi-disciplinary Minor-II Data Visualization, Regression and Prediction		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-II (Data Science)		
Prerequisites: Nil			
Course Outcomes: At the end of the course student will be able to: **CO1.** Demonstrate a strong understanding of core Python concepts. **CO2.** Utilizing powerful Python libraries like Pandas (Data Frames) and NumPy (multidimensional arrays) to load, clean, explore, analyse, and transform datasets of varying sizes and complexities.			

Course Contents		CO Mapped	Teaching Hours/Week
1	**Basics of Python** data types, operators, variables, and expressions. Control structures using sample dataset, objects and functions. Python sequence data structures including String, Array, List, Tuple, Set, and Dictionary, hashes.	1	4
2	**Data Analysis libraries** Using Pandas, Data Frames, Numpy multidimensional arrays, and SciPy libraries to work with a various dataset. Machine Learning in detail.	1	4
3	**Exploratory data analysis** Data pre-processing (data loading, dealing with missing values and outliers, data wrangling). ANN,SVM,KNN,DT etc.	2	4
4	**Exploratory data analysis** Data pre-processing (Data Normalization, Data Formatting, data cleaning). More APIs and library relevant to Data Processing with examples.	2	4

Text Books	
1	Data Visualization with Python and JavaScript, Kyran Dale, Shroff Publisher/O'Reilly Publisher Publication
2	Data Science Using Python and R by Chantal D. Larose and Daniel T. Larose, Wiley Publication.
Recommended Reading	
1	Python for Data Science and Visualization –Beginners to Pro, Udemy.

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2206T		
Course Title	Multi-disciplinary Minor-II Cybersecurity Law & Privacy		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-II (Cyber Security)		
Prerequisites: Nil			

Course Outcomes:
At the end of the course student will be able:
CO1. To be conversant with the social and Intellectual Property issues emerging from Cyberspace.
CO2. To explore the legal and Policy developments in various countries to regulate Cyberspace.
CO3. To develop the understanding of relationship between commerce and cyberspace.
CO4. To gain in-depth knowledge of the Information Technology Act and legal framework of Right to Privacy, Data Security and Data Protection.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Emergence of Cyberspace** Cyber Jurisprudence, Jurisprudence and law, Doctrinal approach, Consensual approach, Real Approach, Cyber Ethics, Cyber Jurisdiction, Hierarchy of courts, Civil and criminal jurisdictions, Cyberspace-Web space, Web hosting and web Development agreement, Legal and Technological Significance of domain Names, Internet as a tool for global access.	1	4
2	**Overview** of IT Act 2000, Amendments and Limitations of IT Act, Analysis of privacy laws such as GDPR, CCPA, HIPAA, Digital Signatures, Cryptographic Algorithm, Public Cryptography, Private Cryptography, Electronic Governance, Legal Recognition of Electronic Records, Legal Recognition of Digital Signature Certifying Authorities, Cyber Crime and Offences, Network Service Providers Liability, Cyber Regulations Appellate Tribunal, Penalties and Adjudication.	2	4
3	**Patent** Law, Trademark Law, Copyright, Software – Copyright or Patented, Domain Names and Copyright disputes, Electronic Database and its Protection, IT Act and Civil Procedure Code, IT Act and Criminal Procedural Code, Relevant Sections of Indian Evidence Act, Relevant Sections of Bankers Book Evidence Act, Relevant Sections of Indian Penal Code, Relevant Sections of Reserve Bank of India Act, Law Relating To Employees And Internet, Alternative Dispute Resolution , Online Dispute Resolution (ODR).	3	4
4	**Evolution and development** in E-commerce, paper vs paperless contracts E-Commerce models- B2B, B2C, E security. Application area: Business, taxation, electronic payments, supply chain, EDI, E-markets, Emerging Trends.	3	4
5	**Case Study** On Cyber Crimes: Harassment Via E-Mails, Email Spoofing (Online A Method Of Sending E-Mail Using A False Name Or E-Mail Address To Make It Appear That The E-Mail Comes From Somebody Other Than The True Sender, Cyber Pornography, Cyber-Stalking.	4	4

6	**Regulatory Compliance** in Cybersecurity, Compliance frameworks: NIST, ISO/IEC 27001, PCI DSS, etc., Regulatory requirements for specific industries (e.g., finance, healthcare), Auditing and certification processes for cybersecurity compliance.	3,4	4

Text Books	
1	K.Kumar," Cyber Laws: Intellectual property & E Commerce, Security", 1st Edition, Dominant Publisher, 2011.
2	Rodney D. Ryder, "Guide to Cyber Laws", Second Edition, Wadhwa And Company, New Delhi, 2007.
Recommended Reading	
1	Information Security policy & implementation Issues, NIIT, PHI.
2	Vakul Sharma, "Handbook of Cyber Laws" Macmillan India Ltd, 2nd Edition, PHI, 2003.

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2207T		
Course Title	Multi-disciplinary Minor-II: Foundations of Software Engineering		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-II (Software Engineering)		
Prerequisites: Introduction to Computer & Software			
Course Outcomes: At the end of the course student will be able to: **CO1.** Demonstrate basic knowledge of Software & Software Engineering. **CO2.** Understand the basics of Software Project Management. **CO3.** Identify & manage risks for software development project. **CO4.** Prepare schedule and track it for a software development project.			

Course Contents		CO Mapped	Teaching Hours/Week
1	**Software Characteristics, Applications & Myths:** Introduction to software characteristics, various applications of software, and common myths associated with software engineering.	1	2
2	**Software Engineering: A Layered Technology:** Overview of software engineering as a layered technology including processes, methods, and tools, Generic View of Software Engineering.	1	2
3	**Software Process & Software Process Models:** Detailed study of software processes and various software process models such as the Prototyping Model, RAD model, Spiral Model, The W^5HH Principle.		3
4	**Software Project Management Concepts:** Introduction to software project management, People, Product, Project, Team Leaders, including planning, execution, and monitoring.	2	3
5	**Software Risk Management:** Reactive Vs. Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring and Management, The RMMM Plan.	3	3
6	**Software Project Scheduling:** Basic Concepts, The Relationship between People and Effort, Defining a Task set for the Software Project, Selecting Software Engineering Task, Refinement of Major Tasks, Defining a Task Network, Scheduling, Earned Value Analysis, Error Tracking, and The Project Plan.	4	3

Text Books	
1	Roger Pressman. Software Engineering: A Practitioner's Approach (5^{th} ed.). McGraw-Hill, Inc., USA. 2000.
2	Ian Sommerville. Software Engineering (9^{th} Edition). Addison-Wesley, 2010.
Recommended Reading	

1	Pankaj Jalote. An Integrated Approach to Software Engineering (3rd Edition). Springer, 2005.

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2208T		
Course Title	Multi-disciplinary Minor-II **Data Structure**		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Multi-disciplinary Minor-II (Advance Computing)		
Prerequisites: Nil			
Course Outcomes: At the end of the course student will be able to: **CO1.** Fundamental data structures (arrays, stacks, queues, linked lists, trees, graphs) and their implementations. **CO2.** Proficient in performing advanced operations on complex data structures such as circular queues, various forms of linked lists, and different types of trees.			

Course Contents		CO Mapped	Teaching Hours/Week
1	Algorithms and Flowcharts, Basics Analysis on Algorithm, Complexity of Algorithm. Introduction to Arrays, Definition, One Dimensional Array and Multidimensional Arrays, Pointer, Pointer to Structure, various Programs for Array and Pointer.	1	6
2	Introduction to Stack, Definition, Stack Implementation, Operations of Stack. Implementation of Multiple Stack Queues. Introduction to Queue, Definition, Queue Implementation, Operations of Queue, Circular Queue.	1	6
3	Introduction, Representation and Operations of Linked Lists, Singly Linked List, Doubly Linked List, Circular Linked List, And Circular Doubly Linked List.	2	6
4	Introduction to Tree, Tree Terminology Binary Tree, Binary Search Tree, Strictly Binary Tree, Complete Binary Tree, Tree Traversal. Introduction, Representation to Graphs, Graph Traversals Shortest Path Algorithms.	2	6

Text Books	
1	Ellis Horowitz, Sartaj Sahni and Dinesh Mehta, “Fundamental of Data Structure using C++”, Galgotia Publication.
2	Y. Langsam, M. J. Augenstein and A. M. Tannenbaum, "Data Structures Using C and C++",, Prentice 2 Hall India, Second Edition.
Recommended Reading	

1	"Data Structures and Algorithms in Python" by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser.
2	"Data Structures and Algorithms Made Easy" by Narasimha Karumanchi

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2007L		
Course Title	Artificial Intelligence Lab		
Course Scheme	**(L-T-P: 0-0-2) (ISCE:60, 40)**		
Credits	1		
Course Type	Program Core Course		
Prerequisites: Nil			
Course Outcomes: At the end of the course student will be able: **CO1.** To compare different search techniques. **CO2.** To apply AI technique for a given problem **CO3.** To implement AI based search technique based on problem. **CO4.** To design AI applications in different domain.			

List of Experiments		CO Mapped	Teaching Hours/Week
1	Implement Breadth first and depth first search techniques.	1	2
2	Implement Hill climbing algorithm.	1	2
3	Implement Beam Search.	2	2
4	Implement A* algorithm.	2	2
5	Implement MINIMAX algorithm.	3	2
6	Implement Alpha Beta algorithm.	3	2
7	Implement AO* algorithm.	4	2
8	Design a smart home application of controlling light using smart speakers such as Amazon Alexa dot ,SIRI Microsoft Cortana etc.	4	2

Text Books	
1	Deepak Khemani,"A First Course in Artificial Intelligence", McGraw Hill Education (India), 2013.
2	Stuart Russell, Peter Norvig, "Artificial Intelligence A Modern Approach", Prentice Hall, 3rd Edition, 2009.
Recommended Reading	
1	Stefan Edel Kamp and Stefan Schroedl, "Heuristic Search: Theory and Application", Morgan Kaufmann, 2011.

2	Zbigniew Michalewicz and David B. Fogel, " How to Solve it: Modern Heuristics",Springer, 2nd Edition, revised and extended edition, 2013

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO2008L		
Course Title	Database Management System Laboratory		
Course Scheme	**(L-T-P: 0-0-2) (ISCE:60, 40)**		
Credits	1		
Course Type	Program Core Course		
Prerequisites: Nil			

Course Outcomes:

At the end of the course student will be able to:

CO1. Apply knowledge to design and implement database tables and views, including creating schemas, defining tables, views, and establishing relationships.

CO2. Develop proficiency in writing SQL queries to retrieve, manipulate, update data stored in databases and perform miscellaneous operations.

CO3. Apply database design principles to create well-structured database schemas ensuring data integrity and efficiency.

CO4. Learn techniques for performing data analysis and generating reports using SQL queries, including aggregation functions, grouping, and filtering.

CO5. Apply PL/SQL database operations.

CO6. Develop critical thinking and problem-solving skills by analysing database requirements for a case study, designing appropriate solutions, and implementing efficient database structures and operations using relational and non-relational data models.

List of Experiments		CO Mapped	Teaching Hours/Week
1	A. Write SQL queries to create database tables, views and establish relationship between them and retrieve, manipulate, and update data stored in databases. B. Study of Joins in SQL.	1	4
2	A. Write SQL queries to retrieve, manipulate, and update data stored in databases. B. Write SQL queries to implement various arithmetic, relational, logical and miscellaneous operators such as IN, NOT IN, BETWEEN, LIKE, EXISTS, ANY, SOME etc on data stored in databases.	2	4
3	Write SQL queries to define integrity constraints on views such as key constraints, entity integrity constraint, referential integrity constraint, domain constraint, UNIQUE, CHECK, NOT NULL, INDEX etc.	3	2
4	A. Write SQL queries to study aggregate functions such as SUM, COUNT, MIN, MAX, AVERAGE, Group By, Order by, HAVING clauses, Built-in String Functions , Data Control Language Commands such as GRANT, REVOKE and Transaction Control Language Commands such as Commit and Rollback. B. Installation of Power BI tool and generating reports for data analysis.	4	4

5	Write PL-SQL queries to define functions, stored procedures, triggers and ranks with partition on data stored in database.	5	4
6	Create an Instagram .sql file with multiple tables and relations and perform various SQL queries to extract the data from the database tables using SQL and NoSQL queries (i.e. MongoDB CRUD operations).	6	3

Text Books	
1	Elmasri & Navathe , "Fundamentals of Database System", 7thEdition, Addison Wesely Publication (2015).
2	Abraham Silberschatz, Henry Korth, Sudarshan , "Database System Concepts", 6th Edition, (2010).
3	Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill, 2002.
Recommended Reading	
1	Michael Mannino, "Database design, Application Development and Administration", ,4th Edition (2008).
2	Peter Rob and Coronel, "Database systems, Design, Implementation and Management", 5th Edition, Thomson Learning, 2001.
3	C. J. Date, "Introduction To Database Systems", Seventh Edition, Addison Wesley Longman.

Programme Name	B. Tech in Computer Engineering	**Semester**	IV
Course Code	R5CO2009L		
Course Title	Software Engineering Laboratory		
Course Scheme	**(L-T-P: 0-0-2) (ISCE:60, 40)**		
Credit	**1**		
Course Type	Program Core Course		
Prerequisites	Computer & Software Fundamentals Laboratory		

COURSE OUTCOMES:

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

CO1: Carry out different umbrella activities of software development project such as cost & time estimation.

CO2: Identify issues in risk management, project scheduling & tracking.

CO3: Assure software quality & Manage software configuration.

CO4: Carry out different framework activities of software development project such as requirements gathering, analysis, design, coding and testing.

LIST OF EXPERIMENTS:		HRS	CO
1	Give detailed Problem Statement	1	1
2	Prepare Software scope	1	1
3	Estimate required Resources	1	1
4	Perform Software cost and time estimation	1	1
5	Perform Risk Analysis and prepare RMMM plan for the case	2	2
6	Prepare Project Schedule	1	2

7	Prepare Software Quality Assurance Plan (SQA plan)	2	3
8	Prepare Project Plan	1	3
9	Carry out Requirement Analysis Modeling	3	4
10	Prepare Software Requirements Specification (SRS)	1	4
11	Carry out Software Design (Data design, Architecture design, Interface design & Component level design)	5	4
12	Write code and implement the software	3	4
13	Develop test cases and carry out testing of the software	3	4
14	Change specifications and make different versions of the software	2	3

Text Books:	
1	Roger Pressman, “Software Engineering”, McGraw Hill, eighth Edition.
2	Ian Somerville, “Software Engineering”, Pearson Education. Sixth Edition.
Reference Books:	
1	W. S. Jawadekar, “Software Engineering”, TMH. 1st Edition
2	R. Mall, “Fundamentals of Software Engineering”, Prentice Hall of India, 2nd Edition

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CO20010L		
Course Title	Linux administration Laboratory		
Course Scheme	**(L-T-P: 0-0-2) (ISCE:60, 40)**		
Credits	1		
Course Type	Vocational and Skill Enhancement Courses		
Prerequisites: Nil			
Course Outcomes: At the end of the course student will be able to: **CO1.** Apply Linux based systems commands. **CO2.** Apply the fundamentals of security in programs, operating systems and databases Demonstrate various Linux distributions File system administration. **CO3.** Identify various issues in Network Administration, Server administration. **CO4.** Implement and configure a server with security administration policy			

List of Experiments		CO Mapped	Teaching Hours/Week
1	Partitioning and Installation: Install two different operating systems (e.g., Linux and Windows) on a dual-boot system, exploring different partitioning schemes.	1	1
2	Shell Scripting: Write a Bash script to automate user account creation, including setting up home directories, default shell, and permissions.	1	1
3	Process Management: Monitor system processes using top, ps, and kill commands. Analyze process behavior under different load conditions.	1	1
4	File System Comparison: Compare the performance and features of different file systems (ext2, ext3, ext4, NTFS, FAT32) using benchmarking tools.	2	1
5	Logical Volume Management (LVM): Create LVM volumes, snapshots, and extend file systems without downtime.	2	1
6	Quotas: Implement disk quotas for users and groups to manage disk space usage.	2	1
7	File System Repair: Create corrupted file systems and use fsck to repair them.	3	1
8	Network File Systems (NFS): Set up an NFS server and client, exploring different mount options and performance optimization.	3	1
9	LAN Setup: Configure a small LAN with multiple devices, including IP addressing, subnet masking, and routing.	3	1
10	DHCP Server: Implement a DHCP server to automate IP address assignment.	4	1
11	DNS Server: Set up a basic DNS server with zone files and name resolution.	4	1
12	Email Server: Configure a mail server (e.g., Postfix) with basic email services.	4	1

13	Network Security: Implement basic firewall rules and network intrusion detection using tools like iptables or firewalld.	4	1

Text Books	
1	Evi Nemeth, Garth Snyder, Ben Whaley, and Trent R. Hein, UNIX and Linux System Administration Handbook, Publisher: Pearson Education; 4th edition, 2010.
2	Wale Soyinka, Linux Administration: A Beginner's Guide, McGraw-Hill Osborne Media Publication, 6thEdition, 2012.
Recommended Reading	
1	Richard Petersen, Linux: The Complete Reference, Shroff/o/'Reilly, McGraw-Hill Education; 6th edition, 2007..
2	Arnold Robbins, Nelson H. F. Beebe, Classic Shell Scripting, 1st edition, 1999.
3	Richard Blum and Christine Bresnahan, Linux Command Line and Shell Scripting Bible, Wiley India Pvt. Ltd.; 2nd edition, 2011.

Programme Name	B. Tech. Computer Engineering	**Semester**	IV
Course Code	R5CH2402O		
Course Title	Environmental Science		
Course Scheme	**(L-T-P: 2-0-0) (TA-MST-ESE : 20-30-50)**		
Credits	2		
Course Type	Value Education Course		
Prerequisites:			

Course Outcomes:

At the end of the course student will be able to:

CO1. Imply the basic knowledge of environmental protection, sustainable development and improvement.

CO2. Categorize and scrutinize impact of human development on natural resources and its conservation.

CO3. Interpret the impact of environmental problems on socio economic growth.

CO4. Apply different Science and Technology (S&T) based sustainability solutions and technological improvement, and methods for the remediation of degraded environment.

CO5. Familiarize with the legislation, management and protocols existing for environmental protection.

Course Contents		CO Mapped	Teaching Hours/Week
1	**Significance of Environment Science** Definition, basic principles and scope of environment science. Need for awareness Industrialization & Urbanization; Basic Ecological Concepts Ecosystems, nature of environmental threats, Current environmental problems, Importance of clean air.	1	6
2	**Natural Resources Management and Sustainability** Concept of Ecosystem, Conservation of ecosystem: Natural Resources, Renewable and Non-renewable Resources, Natural resources and challenges with the conservation. Forest resources, Water resources, Energy resources Role of an individual in conservation of natural resources. Impact of energy use on Environment. Energy conservation and sustainability.	2	8
3	**Environment & Society** Urbanization and environment, social movements, Community participation, JFM, participation by NGOs Impact of energy use on Environment, energy production on environment change, nuclear explosion, impact of dam construction, Energy conservation and sustainability.	3	6
4	**Green Technologies** Role of advancements in science and technology in developing environment friendly technologies 3 R's for Green Technology, Green technology towards sustainable future, Reduction of ecological footprint, Concept of Sustainability and Green Chemistry as a tool for sustainable development.	4	6

5	Environmental Legislation, Management & Policies Aims and Objectives of Environmental Impact Assessment (EIA), Environmental Management Plan (EMP) , Indian forest act, The water act(prevention and control of water pollution), The Air act (prevention & control of air pollution) International efforts for environmental protection and contribution of India for same, National Action Plan on Climate Change Role of Ministry of Environment, forest and climate Mitigation measures for climate change, international protocols, Montreal protocol, Kyoto protocol, Carbon credits and carbon trading.	5	6

Text Books	
1	De., Environmental Chemistry, 6th Edition, New Age International.
2	Erach Bharucha, Text Book of Environmental Studies for Undergraduate Courses, Universities Press, Second Edition (UGC Recommended)
3	P.K.Goel, Water Pollution, Causes, Effects and Control, New Age International
4	Dr. Jagdish Krishnaswamy and Dr. R. J. Ranjit Daniels, Environmental Studies, Wiley India Private Limited, New Delhi, First Edition, 2009.
Recommended Reading	
1	Bharucha Erach, The Biodiversity of India, Mapin Publishing Pvt. Ltd., Ahmedabad,
2	Cunningham, W.P. Cooper, T.H. Gorhani, E & Hepworth, M.T., Environmental Encyclopedia, Jaico Publ. House, Mumbai, 2000
3	Jadhav, H &Bhosale, V.M., Environmental Protection and Laws. Himalaya Pub. House, Delhi, 1995
4	Wanger K.D., Environmental Management. W.B. Saunders Co. Philadelphia, USA, 1998
5	Trivedi R.K., Handbook of Environmental Laws, Rules Guidelines, Compliances and Stadards, Vol I and II, Enviro Media (R)