S.E Syllabus

Semester-III

Theme: **"Tech4Startup: Enabling Rural Entrepreneurship through Digital Tools in Maharashtra"**

Aligned with SDG 4 (Quality Education) and **SDG 9 (Industry, Innovation & Infrastructure).**

Keywords: rural entrepreneurs, information gap, challenges, digital tools.

The theme "Tech4Startup: Enabling Rural Entrepreneurship through Digital Tools" in rural Maharashtra is a critical and multi-faceted area of focus, aiming to leverage technology for the economic empowerment and sustainable development of rural communities.

The theme directly aligns with and contributes significantly to the achievement of UN Sustainable Development Goal (SDG) 4: Quality Education, and SDG 9: Industry, Innovation & Infrastructure, within the specific context of rural Maharashtra.

Alignment with UNSDG 4: Quality Education

SDG 4 aims to "ensure inclusive and equitable quality education and promote lifelong learning opportunities for all." "Tech4Startup" contributes to this by:

- **Digital Literacy and Skills Training:** For rural entrepreneurship to thrive, individuals need foundational digital literacy. This theme implies a focus on providing accessible and relevant training programs to rural communities in Maharashtra. This includes teaching them how to use smartphones, access the internet, navigate online platforms, use digital payment systems (like UPI), and understand basic e-commerce functionalities. This direct digital education empowers aspiring entrepreneurs with the fundamental knowledge to leverage technology for their businesses.
- **Business Skill Development through Digital Platforms:** Beyond basic literacy, "Tech4Startup" involves using digital tools to deliver specialized business education. This could include online courses, webinars, and digital modules on topics such as business planning, marketing (especially digital marketing and social media), financial management, supply chain optimization, and customer relationship management. This ensures that rural entrepreneurs gain the necessary skills to establish and grow sustainable ventures, accessible regardless of their geographical location.
- **Access to Information and Mentorship:** Digital tools facilitate access to a vast repository of information relevant to various industries and markets. Rural entrepreneurs can research market trends, identify gaps, learn about best practices, and connect with mentors or experts from urban centers or even globally. Online forums, professional networking platforms, and dedicated mentorship programs delivered digitally can bridge the knowledge gap and foster a culture of continuous learning.

- **Removing Geographical Barriers to Learning:** Traditional educational infrastructure might be scarce in remote rural areas. Digital tools effectively overcome these geographical barriers, making quality education and skill development accessible to a wider audience, including women, youth, and other marginalized groups who may have limited mobility.

Alignment with UNSDG 9: Industry, Innovation & Infrastructure

SDG 9 focuses on "building resilient infrastructure, promoting inclusive and sustainable industrialization and fostering innovation." "Tech4Startup" directly contributes to this by:

- **Developing Digital Infrastructure:** The success of "Tech4Startup" hinges on robust digital infrastructure in rural Maharashtra. This includes ensuring reliable internet connectivity (broadband, mobile data), access to affordable digital devices, and the establishment of community digital centers. Initiatives like the government's BharatNet project are crucial to laying this foundation, enabling rural entrepreneurs to connect to the global digital economy.
- **Fostering Innovation in Rural Industries:** Digital tools empower rural entrepreneurs to innovate in their products, processes, and business models. This could involve using data analytics to understand market demand, adopting new agricultural technologies (Agri-Tech), implementing efficient production techniques, or creating unique digital marketing strategies for local products. It encourages a shift from traditional practices to more efficient and competitive approaches.
- **Promoting Inclusive Industrialization:** By enabling rural entrepreneurship, the theme contributes to inclusive industrialization. It means empowering small-scale and cottage industries, handicrafts, agro-processing units, and service-based businesses in rural areas to scale up and reach wider markets. Digital platforms (e-commerce, social media marketing) are vital for these businesses to overcome limitations of physical storefronts and localized demand.
- **Creating Resilient Business Ecosystems:** Digital tools enhance the resilience of rural businesses by diversifying market access (online sales reduce reliance on local markets), streamlining operations, and improving financial inclusion (digital payments, access to credit). It allows entrepreneurs to adapt to changing market conditions and economic shocks more effectively.
- **Stimulating Local Economic Growth and Job Creation:** Rural startups, powered by digital tools, can create employment opportunities within their communities, reducing rural-urban migration. By fostering a vibrant entrepreneurial ecosystem, the theme encourages local value creation, leading to sustainable economic growth in Maharashtra's rural regions. This also attracts investments and develops ancillary services.

Problem Statements:

1. **“Significant information gap faced by rural entrepreneurs in Maharashtra regarding crucial government schemes and the practical benefits of digital tools, ensuring this information is delivered intuitively and culturally relevantly in local languages.”**
2. **“Inefficiencies and high costs associated with last-mile delivery and pickup of goods for rural entrepreneurs in remote areas of Maharashtra, given the fragmented nature of local transport options.”**
3. **“The challenges faced by rural entrepreneurs in Maharashtra in effectively showcasing their unique products and services, establishing credible branding, ensuring quality assurance, and managing last-mile logistics to access broader national and international markets.”**
4. **"There is no real-time data-driven support system to help rural entrepreneurs in Maharashtra make informed decisions on production, pricing, and distribution."**
5. **"Digital content and training materials relevant to rural entrepreneurship are often not available or easily understandable in local languages and dialects prevalent across Maharashtra's diverse rural communities."**
6. **"Women in rural areas of Maharashtra face barriers in accessing digital tools for entrepreneurship due to societal norms and lack of tailored training."**

R-2025 S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per week)			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12211301	**Discrete Mathematics and Graph Theory**	**2**	**1**	**--**	**30**	**15**	**--**	**45**	**90**	**3**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Semester Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+ IAT-2					
12211301	**Discrete Mathematics and Graph Theory**	**20**	**20**	**40**	**60**	**2.5**	**25**	**--**	**125**

Rationale:

Elementary Number Theory provides the mathematical foundation for cryptography and security algorithms essential in computer engineering, such as RSA encryption. Functions and Mathematical Relations help model data structures, database schemas, and finite state machines used in software and hardware design. The Fundamental Principle of Counting equips students with combinatorial techniques crucial for analyzing algorithms, calculating IP addressing schemes, and designing logic circuits. Algebraic Structures like groups, rings, and fields underpin error detection, hashing functions, and cryptographic protocols. Graph Theory enables modeling of networks, dependency graphs, and routing algorithms vital for computer networks and compiler design. Finally, Coding Theory teaches methods for error detection and correction, enhancing data

reliability in communication systems and memory storage. Together, these topics build a rigorous mathematical framework for solving practical problems in computer engineering.

Course Objectives:

1. To understand number theory concepts such as divisibility, primes, and modular arithmetic.

2. To explain types of functions and relations and their properties for use in system modeling.

3. To describe and apply combinatorial techniques to solve basic counting problems.

4. To understand algebraic structures and their applications in computing.

5. To explain fundamental graph theory concepts and their role in modeling networks.

6. To understand principles of error detection and correction using coding theory.

Course Outcomes:

1. Students will be able to apply modular arithmetic to solve problems like cryptographic key generation in RSA encryption.

2. Students will be able to analyze function mappings and relations in database schema design and finite automata transitions.

3. Students will be able to solve real-world problems such as determining the number of IP address allocations or password combinations.

4. Students will be able to use group and field theory to design reliable hashing algorithms and error-detecting codes.

5. Students will be able to model and analyze computer networks, routing algorithms, and dependency graphs using graph theory.

6. Students will be able to apply Hamming codes and parity checks to enhance data reliability in memory systems and communication protocols.

Prerequisite:

1. Definition of sets, Venn diagrams, complements, cartesian products, power sets.
2. Propositions and Logical operations, Truth tables, Equivalence, Implications

R-2025 S.E Computer Engineering

DETAILED SYLLABUS

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	Elementary Number Theory	Modular Arithmetic, Divisibility and Euclid Algorithm,	05	CO1
		Primes and Sieve of Eratosthenes.		
		Euler's and Fermat Little Theorem, Congruences, Computing Inverse in Congruences		
		Legendre and Jacobi Symbol, Chinese Remainder Theorem and Well-ordering principle.		
		Self-learning Topics: 1. Wilson's Theorem 2. Applications in modular exponentiation. 3. Quadratic residues 4. Law of Quadratic Reciprocity 5. Continued fractions and Diophantine equations 6. Primitive roots and orders mod n.	10	
II	Functions and Mathematical relations	Types: Injective, Surjective, and Bijective Functions. Composition, Inverse Functions. Real life applications of Functions.	05	CO2
		Partial Order Relations, Poset, Hasse Diagram, Chain and Anti chains, Lattice, Types of Lattices, Sub lattice.		
		Self-learning Topics: 1. Practical applications of function in Neural Network. 2. Determining risk factors for insurance rates, Taxes and tax brackets, Vending machines, etc. 3. Practical applications of relations in real life in the field of Database Management, Economics, Social Network, Sports, Medical Diagnosis, Weather, etc. 4. Equivalence Classes.	10	
		Basic Counting Principle-Sum Rule, Product Rule.		

III	Fundamental Principle of Counting	Inclusion-Exclusion Principle and Pigeonhole Principle	05	CO3
		Recurrence relations, Solving recurrence relations		
		Self-learning Topics: 1. Applications of Recurrence Relations – Analysis of recursive algorithms in computing. 2. Combinatorial Problem Solving – Using counting techniques in probability and decision-making.	10	
IV	Algebraic Structures	Algebraic structures with one binary operation: Semi group, Monoid,	05	CO4
		Groups and Subgroups.		
		Abelian Group, and Cyclic group.		
		Self-learning Topics: 1. Algebraic structures with two binary operations: Ring. 2. Integral domain 3. Finite fields. 4. Error Correcting codes. 5. Isomorphism of groups and rings.	10	
V	Graph Theory	Types of graphs, Graph Representation, Sub graphs.	05	CO5
		Operations on Graphs, Walk, Path, Circuit, Connected Graphs, Disconnected Graph, Components of Graph.		
		Homomorphism and Isomorphism of Graphs.		
		Euler and Hamiltonian Graphs, Planar Graph, Cut Set, Cut Vertex.		
		Trees and application of trees.		
		Self-learning Topics:	10	

		1. Network Flow Problems – Understanding flow in networks and its optimization. 2. Graph Coloring Applications in Scheduling – Use of graph coloring in timetabling and resource allocation. 3. Optimization Techniques – Application of graphs in shortest path problems, spanning trees, and clustering. 4. Spanning trees.		
VI	Coding Theory	Introduction to coding	05	CO6
		Group Codes, Decoding and Error Correction		
		Maximum Likelihood Decoding Technique		
		Self-learning Topics: 1. Cyclic and Polynomial Codes 2. Linear Block Codes 3. BCH and Reed–Solomon Codes	10	

Text Books:

1. Susanna S. Epp, “Discrete Mathematics with Applications”, 5th Edition, Cengage Publications.

2. Ralph P. Grimaldi, “Discrete and Combinatorial Mathematics”, 5th Edition, Pearson Publications.

3. Edgar Goodaire and Michael Parmenter, “Discrete Mathematics and Graph Theory”, 3rd Edition, Pearson Publications.

Reference Books:

1. Kenneth A. Ross, “Discrete Mathematics”, 5th Edition, Pearson Publications.

2. Swapan Kumar Sarkar, “Textbook of Discrete Mathematics”, 9th Edition, S. Chand Publications.

3. Bernad Kolman, Robert Busby, Sharon Cutler Ross, Nadeem-ur-Rehman, “Discrete Mathematical Structures”, 6th Edition, Pearson Education.

4. T. Veera Rajan, "Discrete mathematics with Graph Theory and Combinatorics", McGraw Hill Publications.

5. C. L. Liu "Elements of Discrete Mathematics", second edition 1985, McGraw-Hill Book Company.

Online References:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106106094
2.	https://nptel.ac.in/courses/106108227
3.	https://nptel.ac.in/courses/106106183
4.	https://nptel.ac.in/courses/106103205
5.	https://nptel.ac.in/courses/111107058
6.	https://nptel.ac.in/courses/106103015

Term work (TW) for 25 marks:

General Instructions:

1. Batch wise tutorials are to be conducted. The number of students per batch should be as per the university pattern for practical.
2. Students must be encouraged to write **at least 10** class tutorials on entire syllabus.
3. A group of 4-6 students should be assigned a self-learning topic. Students should prepare a presentation/problem solving of 10-15 minutes. This should be considered as mini project.

This project should be graded for 10 marks depending on the performance of the students.

The distribution of Term Work marks will be as follows –

1. Regularity and active involvement (Theory and Tutorial) 05 marks.

2. Class Tutorials on entire syllabus 10 marks.

3. Mini Project 10 marks.

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12211302	Analysis of Algorithm	3	--	--	45	--	--	45	90	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+IAT-2					
12211302	Analysis of Algorithm	20	20	40	60	2.5	--	--	100

R-2025- S.E Computer Engineering

Rationale:

The Analysis of Algorithm syllabus equips students with essential algorithmic skills to solve real-world problems efficiently. Aligned with the theme ***"Tech4Startup: Enabling Rural Entrepreneurship through Digital Tools in Maharashtra"***, it emphasizes practical applications like rural logistics, scheduling, and search systems. Through modules on Greedy, Dynamic, and Graph algorithms, students learn to build digital tools that support rural innovation. The course fosters a problem-solving mindset aligned with **SDG 4 and SDG 9**.

Course Objectives:

1. Awareness of how algorithm performance affects time and space efficiency.
2. Exposure to Divide and Conquer techniques for solving common computational problems.
3. Creation of logic for solving rural logistics and task scheduling using Greedy methods.
4. Understanding of Dynamic Programming concepts for building planning and support tools.
5. Usage of Backtracking and Branch & Bound methods for scheduling and resource allocation in rural applications.
6. Application of string matching techniques for building simple, local-language search functions in digital tools.

Course Outcomes:

1. **Analyze** algorithm performance using time and space complexity.
2. **Implement** Divide and Conquer methods to improve rural data processing.
3. **Design** Greedy algorithms for optimizing rural logistics and scheduling.
4. **Develop** Dynamic Programming solutions for rural planning tasks.

5. **Apply** Backtracking and Branch & Bound for scheduling and resource planning.
6. **Use** String Matching to build search features for rural digital apps.

Prerequisite: Data Structure Concepts

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Arrays, Queues, Graphs, Sorting and Searching, Recursion		
I	**Introduction to Algorithms**	Performance analysis- Master Method, space, and time complexity Growth of function, Big- Oh, Omega Theta Notation Mathematical background for algorithm analysis, Recursion tree method, Analysis of selection sort, insertion sort.	**7**	**CO1**
		Self-learning Topics: Complexity class: Definition of P, NP, NP-Hard, NP-Complete, substitution method, **Compare** the time taken by Bubble Sort Vs Insertion Sort for sorting 50 product prices from a local market and Analyze which sorting is better based on input	**7**	

		size.		
II	**Divide and Conquer Approach**	General method, Merge sort, Quick sort, Analysis of Binary search., Finding minimum and maximum algorithms and their Analysis.	**7**	**CO2**
		Self-learning Topics:, Strassen's Algorithm, **Build** a **"Product Search Tool"** where users can search for an item in a sorted price list using Binary Search (Apply recursive problem-solving for fast searching.)	**6**	
III	**Greedy Method Approach**	General Method, Single source shortest path: Dijkstra Algorithm, Fractional Knapsack problem, Minimum cost spanning trees: Kruskal and Prim's algorithms	**8**	**CO3**
		Self-learning Topics: Job sequencing with deadlines, BFS and DFS Graph traversal techniques, **design** a route planner that finds the **shortest path** from farms to nearby villages using Dijkstra's algorithm (Solve basic optimization problems)	**7**	

IV	**Dynamic Programming Approach**	General Method, Multistage graphs, all pair shortest path: Floyd Warshall Algorithm, 0/1 knapsack Problem, Travelling Salesperson problem, Longest common subsequence.	**10**	**CO4**
		Self-learning Topics: Bellman Ford Algorithm, assembly-line scheduling problem, **solve** a **"Grain Storage Optimization"** problem using 0/1 Knapsack – decide which grain bags to load in a truck with limited weight capacity (Practice decision-making using memory-based optimization.).	**10**	
V	**Backtracking and Branch and Bound**	General Method, **Backtracking:** N-queen problem, Sum of subsets, Graph coloring. **Branch and Bound:** Travelling Salesperson Problem, 15 Puzzle problem.	**8**	**CO5**
		Self-learning Topics: Create a **simple class timetable scheduler** using Backtracking, ensuring no teacher is double-booked.	**8**	

		(Understand constraint-based problem-solving.)		
VI	**String Matching Algorithms**	The Naïve string-matching algorithm, The Rabin Karp algorithm, The Knuth-Morris-Pratt algorithm.	**5**	**CO6**
		Self-learning Topics: Build a Marathi/vernacular search tool to help farmers find specific training videos or articles on pest control, using KMP string matching.	**7**	

Text Books:

1. T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms", 2nd Edition, PHI Publication 2005
2. Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms" University Press.

References:

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw- Hill Edition.
2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI

Online References:

Sr. No.	Website Name
1	https://nptel.ac.in/courses/106/106/106106131/
2	https://swayam.gov.in/nd1_noc19_cs47/preview
3	https://www.coursera.org/specializations/algorithms
4	https://www.mooc-list.com/tags/algorithms

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12211303	Database Management System	3	-	-	45	--	--	45	90	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -1	IAT-2	IAT-1+ IAT-2					
12211303	Database Management System	20	20	40	60	2.5	--	--	100

Rationale:

This course introduces students to the fundamentals of database systems essential for building data-driven applications. It equips learners with practical skills in data modeling, SQL, normalization, and modern database technologies. The focus is on designing scalable, reliable solutions to support rural entrepreneurship through digital tools.

Course Objectives:

1. To introduce fundamental database concepts and demonstrate their role in building digital solutions for rural entrepreneurship.
2. To enable students to model data using Entity-Relationship and relational models tailored to rural enterprise needs.
3. To develop skills in Structured Query Language (SQL) for creating, retrieving, and managing data in digital applications.
4. To explain the principles of data normalization and guide students in designing optimized, consistent databases.
5. To impart knowledge of transaction management, concurrency control, and recovery for reliable digital platforms.
6. To expose students to emerging database technologies like NoSQL and Big Data for scalable rural tech solutions.

Course Outcomes:

1. Identify and describe core database concepts relevant to rural digital systems.
2. Design ER and relational models based on rural entrepreneurship case studies.
3. Develop SQL queries to manage and retrieve data for rural enterprise applications.
4. Apply normalization techniques to optimize rural business databases.
5. Analyze transaction handling and concurrency control in rural tech environments.
6. Create scalable database-driven solutions using emerging technologies for rural entrepreneurship.

Prerequisite: Programming Fundamentals, Computer Fundamentals, Discrete Structures

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basics of computer systems, data types, file systems, data vs. information, intro to software tools like MS Excel/Google Sheets	-	
I	**Fundamentals of Data**	Introduction to databases and their applications, Characteristics of databases, DBMS architecture, ACID properties, The	6	CO1

	Modelling & E-R Design	Entity-Relationship (ER) model, Entity types: weak and strong entity sets, Types of attributes and keys, Relationship constraints: cardinality and participation, Extended Entity-Relationship (EER) model: generalization, specialization, aggregation.		
		Self-learning Topics: Design an ER model for a real-time rural startup case study, Map use cases to entities and relationships, Investigate real-world use of databases in rural entrepreneurship platforms.	6	
II	**Relational Model and Relational Algebra**	Relational model concepts, Concepts of keys in relational model, Mapping ER to relational model, Relational algebra operations: select, project, union, set difference, Cartesian product, rename, join operations (theta, equi, natural), Use of relational algebra in query formulation	8	CO2
		Self-learning Topics: Design relational schemas for a rural enterprise information system, Practice relational algebra queries using real-life rural datasets, Compare different join operations using case-based examples, Explore tools that visualize relational algebra execution.	8	
III	**Structured Query Language (SQL)**	Introduction to SQL, DDL, DML, DCL commands, SQL queries for data retrieval, Constraints and integrity rules, Joins and subqueries, Aggregation and grouping, Views and indexing, SQL use-cases for rural data systems like inventory, payments, and orders.	8	CO3
		Self-learning Topics: Write SQL queries for a rural startup's digital backend (customer, product, order), Create views and test query optimization, Design and test constraints for data accuracy, Practice advanced SQL on open-source database tools	8	

IV	**Normalization and Database Design**	Functional dependencies, Types of anomalies, First, Second, Third and Boyce-Codd Normal Forms (1NF, 2NF, 3NF, BCNF), Multi-valued dependencies and 4NF (overview), Database design approach, Case studies for rural entrepreneurship data organization.	7	CO4
		Self-learning Topics: Normalize datasets from a rural entrepreneurship domain, Analyze real-world database structures for redundancy and optimization, Redesign a denormalized rural enterprise schema, Explore online tools for visual database modeling.	7	
V	**Transaction Management and Concurrency Control**	Transaction concepts, States of transactions, Serializability and schedules, Concurrency control protocols (2PL, Timestamp-based), Deadlock handling, Recovery concepts, Use of transactions in digital payment platforms used in rural setups (e.g., UPI, microcredit systems).	8	CO5
		Self-learning Topics: Model concurrent transactions in rural order-processing systems, Simulate conflicts and deadlocks in sample scenarios, Explore open-source database engines with transaction logs, Analyze transaction case studies from fintech and rural startups.	8	
VI	**Introduction to Modern Databases**	NoSQL databases overview (Document, Column, Key-Value stores), Introduction to Big Data and Hadoop ecosystem, Role of databases in AI/ML integration, Case study: database backends in e-commerce and logistics platforms for rural businesses, Data-driven decision making and dashboards.	8	CO6
		Self-learning Topics: Explore NoSQL tools (MongoDB, Firebase) using rural product catalogs, Create mock dashboards for rural entrepreneurs using Power BI or Google Data Studio, Review	8	

		government digital initiatives for rural entrepreneurship (e.g., PMGDISHA), Design a lightweight data management solution for a rural startup.		

Text Books:

1. Elmasri and Navathe, *Fundamentals of Database Systems*, 7th Edition, Pearson Education
2. Korth, Silberchatz, Sudarshan, Database System Concepts, 7th Edition, McGraw Hill
3. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, 3rd Edition, Tata McGraw Hill

References:

1. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing, Tata McGraw Hill Education
2. Pramod J. Sadalage and Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley
3. C.J. Date, An Introduction to Database Systems, 8th Edition, Pearson Education
4. Mark L. Gillenson, Fundamentals of Database Management Systems, Wiley

Online References:

Sr. No.	Website Name
1	NPTEL Lecture Series: Database Management system By Prof. Partha Pratim Das, Prof. Samiran Chattopadhyay | IIT Kharagpur
2	https://www.classcentral.com/course/swayam-database-management-system-9914
3	https://www.mooc-list.com/tags/dbms

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30)
12211304	Object Oriented Programming	3	-	-	45	-	-	45	90	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -1	IAT -2	IAT-1+ IAT-2					
12211304	Object Oriented Programming	20	20	40	60	2.5	-	-	100

Rationale:

The Object Oriented Programming syllabus empowers learners to develop strong programming foundations essential for building digital solutions that can drive **rural entrepreneurship in Maharashtra under the Tech4Startup initiative**. By teaching structured and object-oriented programming concepts, it enables the creation of reliable, maintainable software tools that support small businesses, digital literacy, and automation. The course directly contributes to **SDG 4** by enhancing access to quality technical education, and to **SDG 9** by fostering innovation and promoting technology-based infrastructure development in underserved regions.

Course Objectives:

1. Explore the foundational concepts of Java and code structured programs using data types, operators, control structures, and input/output mechanisms.
2. Construct modular applications by applying object-oriented principles through classes, objects, arrays, and string handling techniques.
3. Design extensible Java programs by applying inheritance, interface implementation, and package structuring to promote code reusability and scalability.
4. Formulate robust strategies for handling runtime errors using Java's exception handling mechanisms to ensure program reliability.
5. Configure concurrent Java applications using multithreading concepts to achieve responsive and efficient program execution.
6. Integrate Swing-based GUI components with JDBC to create interactive and data-driven desktop applications.

Course Outcomes:

1. Students will be able to implement basic Java programs using appropriate syntax, logic, and programming constructs for real-world problem-solving.
2. Students will be able to develop reusable and organized Java applications using constructors, arrays, and string classes.
3. Students will be able to synthesize and deploy Java applications that utilize inheritance hierarchies, interface-based design, and modular code packaging.

4. Students will be able to apply structured exception handling constructs (try, catch, throw, throws, and finally) to manage application errors effectively**.**
5. Students will be able to develop multithreaded Java programs using thread classes, synchronization techniques, and concurrency management tools.
6. Students will be able to construct and demonstrate Java applications that combine user interfaces with backend database operations for real-world use cases.

Prerequisite: Structured Programming Approach or knowledge of any programming language

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Structured Programming Approach or knowledge of any programming language		
I	**Introduction to Java Programming.**	Fundamentals of Java Programming: Overview of procedure and object-oriented programming, Features of Java, Java Virtual Machine Principles of OOP: Object, Class, Encapsulation, Abstraction, Inheritance, Polymorphism Basic Constructs: Constants, variables and data types, Wrapper classes, Operators and Expressions, Type conversion and casting	**6**	**CO1**

		Input & Output in Java: command line arguments, BufferedReader class and Scanner class.		
		Self-learning Topics: Branching and looping: if, if-else, nested if-else, if-else-if ladder, switch-case, break, continue, for loop, while loop, and do-while loop.	**6**	
II	**Classes and Objects, Arrays and Strings**	Access specifiers, static and non-static members, Passing and returning variables and references, Method Overloading Arrays, Array of Objects, Strings (String and StringBuffer classes) Constructors: Default, Parameterized Constructors, copy constructor and Constructor overloading.	**9**	**CO2**
		Self-learning Topics: Heap Memory: Structure, object allocation, object promotion Stack Memory: Function call management, local variables, reference storage, stack overflow Method Area: Class metadata, static data, constant pool, concept of Metaspace	**9**	

		Active Object and Active Reference: Object reachability and reference tracking Inactive Object and Inactive Reference: Conditions for object inactivity, use of null references Garbage Collector: Concept, types (Serial, Parallel, CMS, G1), phases (Mark, Sweep, Compact), GC tuning basics		
III	**Inheritance, Interface and Packages**	Inheritance Basics, Types of Inheritance in Java, Concept of Super and sub class, inheriting Data members and Methods, Role of Constructors in inheritance, making methods and classes final, Method overriding, Dynamic Method Dispatch, Abstract classes and methods. Defining an interface, extending interfaces, implementing interfaces, accessing implementations through interface references, Interfaces vs. Abstract classes Packages – Steps for defining, creating and accessing a Package, importing packages, Making JAR Files for Library Packages, java.util.Vector	**9**	**CO3**

		Self-learning Topics: Constructor Chaining and Object Initialization, Memory Allocation for Inherited Objects, Stack and Heap Usage in Inheritance Contexts, Dynamic Binding and Method Overriding, Garbage Collection in Inherited Class Structures, Making JAR Files for Library Packages.	**9**	
IV	**Exception Handling**	Define Exception, advantages of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, userdefined exception classes (creating own exception subclasses)	**7**	**CO4**
		Self-learning Topics: Impact of Exception Handling on Performance: Performance considerations, benchmarking exception-heavy code. Using Assertions vs. Exceptions: When to use assertions for validation and exceptions for error handling.	**7**	
V	**Multithreading**	Need of Multithreading, Java thread Model, thread Life-Cycle, thread class Methods, Daemon Threads Implementing Runnable, extending thread, synchronizing threads, synchronized Statement, Critical Factor in Thread –Deadlock	**7**	**CO5**

		Self-learning Topics: Java Concurrency, Utilities (java.util.concurrent package), Volatile Keyword and Memory Consistency, Thread Local Storage, Atomic Variables and Locks, Fork/Join Framework, Handling Exceptions in Threads.	**7**	
VI	**GUI programming in JAVA**	SWING Programming: Swing Components and Containers, Swing Packages, A Simple Swing Application, Designing Swing GUI Application and Event handling. JDBC Introduction, JDBC Architecture, Types of JDBC Drivers, The Connectivity Model, The java.sql package, Navigating the ResultSet object's contents, Manipulating records of a ResultSet object through User Interface , The JDBC Exception classes, Database Connectivity, Data Manipulation (using Prepared Statements, Joins, Transactions, Stored Procedures), Data navigation.	**7**	**CO6**
		Self-learning Topics: Application Themes for Self- Development: Health Tracker System (SDG 3: Good Health and Well-being), Student Quiz & Feedback App (SDG 4: Quality Education), Water Usage Logger (SDG 6: Clean Water and	**7**	

		Sanitation) Waste Collection Feedback App (SDG 11: Sustainable Cities), Eco-Friendly Product Inventory System (SDG 12: Responsible Consumption)		

Text Books:

1. Herbert Schildt, "Java-The Complete Reference", Thirteenth Edition, Tata McGraw Hill Publication
2. E. Balguruswamy, "Programming with java A primer", Seventh edition, Tata McGraw Hill Publication
3. Dr.G.T.Thampi, Junaid Khateeb,"Computer Programming in Java", Wiley Publication.

References:

1. D.T. Editorial Services, "Java 8 Programming Black Book", Dreamtech Press
2. H. M.Deitel, P. J. Deitel, S. E. Santry, "Advanced Java 2 Platform How to Program" Prentice Hall
3. Jaime Nino, Frederick A. Hosch, "An introduction to Programming and Object Oriented Design using Java", Wiley Student Edition.

Online References:

Sr. No.	Website Name
1	https://www.w3schools.com/java/
2	https://www.programiz.com/java-programming
3	https://docs.oracle.com/javase/tutorial/

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12212305	Integrated Lab- 1	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-1	IAT-2	IAT-1+ IAT-2				
12212305	Integrated Lab - 1	--	--	--	--	25	25	50

Lab Objectives:

1. Implement and compare the performance of fundamental sorting and searching algorithms.
2. Apply algorithmic techniques like Divide and Conquer, Greedy, and Dynamic

Programming to solve practical problems.

3. Develop solutions for rural network routing and scheduling problems using graph and backtracking algorithms.
4. Design and model databases for rural entrepreneurship scenarios using ER diagrams and normalization.
5. Write and execute SQL queries to manage and analyze data in rural business applications.
6. Explore NoSQL databases and visualization tools to support data management and decision-making in rural enterprises.

Lab Outcomes:

1) **Implement** and **test** classical algorithms (Insertion, Selection, Quick, Merge Sort) and measure their time complexities.
2) **Analyze** and **solve** practical problems using advanced algorithms (Dijkstra's, Longest Common Subsequence, N-Queens) applied to rural contexts.
3) **Design** and **evaluate** graph-based and backtracking solutions for routing and scheduling challenges in rural systems.
4) **Create** ER diagrams and convert them into normalized relational schemas tailored to rural business requirements.
5) **Develop** and **analyze** complex SQL queries for effective data extraction and manipulation in rural enterprise databases.
6) **Evaluate** transaction control, concurrency, NoSQL implementation, and data visualization techniques for enhancing rural business intelligence.

List of Experiments.

Sr No	List of Experiments	Hrs
01	To implement a program for Insertion sort and Selection sort. Also find its complexity.	2
02	To implement Quick Sort and Merge Sort Algorithm using Divide and Conquer and find its complexity.	2
03	To implement Min-Max Algorithm using Divide and Conquer and find its complexity.	2
04	To implement single Source Shortest Path using Dijkstra's algorithm using Greedy Approach.	2
05	Designing Digital Routing Paths in Rural Networks Using BFS and DFS Graph Traversal Algorithms.	2
06	To implement Longest Common Subsequence Algorithm using Dynamic Approach.	2
07	Scheduling Village Workshops or Health Camps Using N-Queens Algorithm to Avoid Double-Booking.	2
08	To implement Knuth Morris Pratt String Matching Algorithm.	2
09	Design an ER diagram for a rural entrepreneurship use case.	2
10	Map an ER model to relational schema and normalize it.	2
11	Implement SQL DDL and DML for a rural business application.	2
12	Write SQL queries to extract and analyze rural business data.	2
13	Simulate transactions with concurrency control in a rural inventory system.	2
14	Create and test a NoSQL database for a rural product catalog.	2
15	Design a data dashboard for a rural startup using Power BI or Google Data Studio.	2

Sr No	List of Assignments / Tutorials	Hrs.
01	Assignment covers the topics from first three modules of Analysis of Algorithms :Introduction to algorithms, Divide & Conquer Approach, Greedy Approach	2
02	Assignment covers the topics from Last three modules of Analysis of Algorithms : Dynamic Programming, Backtracking & Branch and Bound, String Matching	2
03	Assignment covers the topics from first three modules of Database Management System :Fundamentals of Data Modelling & E-R Design, Relational Model and Relational Algebra, Structured Query Language (SQL)	2
04	Assignment covers the topics from last three modules of Database Management System : Normalization and Database Design, Transaction Management and Concurrency Control, Introduction to Modern Databases	2

Text Books:

1. T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms", 2nd Edition, PHI Publication 2005
2. Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms" University Press.
3. Elmasri and Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Education

References:

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw- Hill Edition.
2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI
3. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing, Tata McGraw Hill Education

Online Resources:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106/106/106106131/
2.	https://swayam.gov.in/nd1_noc19_cs47/preview
3.	https://www.coursera.org/specializations/algorithms
4.	https://www.mooc-list.com/tags/algorithms
5.	NPTEL Lecture Series: Database Management system By Prof. Partha Pratim Das, Prof. Samiran Chattopadhyay | IIT Kharagpur

Assessment:

Term Work: Term Work shall consist all practical's based on the above list. Also, Term work Journal must include at least 2 assignments per subject.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits(C) (Notional Learning Hour/30
12212306	**Object Oriented Programming Lab**	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-1	IAT-2	IAT-1+ IAT-2				
12212306	**Object Oriented Programming Lab**	--	--	--	--	25	25	50

Lab Objectives:

1. Introduce basic Java syntax, data types, and standard input/output operations.
2. Develop skills in applying object-oriented programming concepts such as method overloading, inheritance, and packages.
3. Explore exception handling techniques to write robust and error-tolerant programs.
4. Implement multithreading and synchronization concepts for concurrent Java applications.
5. Create interactive user interfaces using Swing components for desktop applications.

6. Demonstrate database connectivity using JDBC for performing basic CRUD operation.

Lab Outcomes:

1. **Identify** and **apply** fundamental Java constructs for input/output and string manipulation.
2. **Construct** object-oriented programs demonstrating inheritance, overloading, and modular design.
3. **Implement** exception handling mechanisms to manage runtime errors effectively.
4. **Develop** multithreaded applications demonstrating thread creation and synchronization.
5. **Build** interactive GUI applications using Swing components.
6. **Perform** database operations using JDBC for data retrieval and manipulation.

List of Experiments:

Sr No	List of Experiments	Hrs
01	Write a Java program to understand how to accept input using Scanner or BufferedReader and print output using System.out.println statement.	2
02	Write a java program to test whether string is palindrome or not	2
03	Write a java program to count number of vowels and consonants from the given strings	2
04	Write a Java program to demonstrate Method overloading	2
05	Write a java program to demonstrate Array and Vectors operations	2
06	Write a java programs to add n strings in a vector array. Input new string and check whether it is present in the vector. If it is present delete it otherwise add it to the vector.	2
07	Write a java programs to demonstrate hierarchical inheritance	2
08	Write a java program to demonstrate Modules and packages	2

09	Write java programs to demonstrate Exception handling using try, catch, throw, throws and finally statements.	2
10	Write java program to create a user defined Exception class known as PayOutOfBoundsException	2
11	Write java programs to create user defined threads by extending thread class and by implementing runnable.	2
12	Write java program to implement the concept of Thread Synchronization	2
13	Write a Java program to implement Swing components namely Buttons, ,JLabels, Checkboxes, Radio Buttons, JScrollPane, JList, JComboBox, Trees, Tables Scroll pane Menus and Toolbars to design interactive GUI.	2
14	Write a Java program using JDBC to connect to a MySQL (or SQLite) database, retrieve all records from a simple table (e.g., students with fields like id, name, marks), and display the records in the console.	2
15	Write a Java program that connects to the same database and performs: Insert a new student record into the students table using PreparedStatement, Update the marks of an existing student identified by ID.	2

Sr No	List of Assignments / Tutorials	Hrs
01	Assignment covers the topics from first three modules of Object Oriented Programming Language: Introduction to Java Programming, Classes and Objects, Arrays and Strings, Inheritance, Interface and Packages	2
02	Assignment covers the topics from last three modules of Object Oriented Programming Language: Exception Handling, Multithreading , GUI Programming in JAVA.	2

Text Books:

1. Herbert Schildt, *"Java - The Complete Reference"*, **Thirteenth Edition**, McGraw

Hill Publication, 2023.
2. E. Balaguruswamy, "Programming with Java: A Primer", Seventh Edition, Tata McGraw Hill Publication (2023).
3. Star EduSolutions, "Learn to Master Java", Paperback, 1st Edition, Star EduSolutions Publication.

References:

1. Java 8 Programming Black Book", Dreamtech Press, 1st Edition (2015).
2. H. M. Deitel, P. J. Deitel, S. E. Santry, "Advanced Java 2 Platform How to Program", Prentice Hall, 1st Edition (September 28, 2001)
3. Jaime Niño, Frederick A. Hosch, "An Introduction to Programming and Object-Oriented Design Using Java", Wiley Student Edition, 3rd Edition (February 4, 2008).

Online Resources:

Sr. No.	Website Name
1	https://www.codestepbystep.com/problem/list/java
2	https://www.programiz.com/java-programming
3	https://www.includehelp.com/java-programs.aspx

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

R-2025 SE Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours)			Teaching Scheme (Contact Hours Per Semester)					Total
		L	T	P	L	T	P	SL	Notional Learning Hour	
9951137WW	**Open Elective**	**2**	**-**	**-**	**30**	**-**	**-**	**30**	**60**	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT 1	IAT 2	IAT-1+IAT-2					
9951137WW	**Open Elective**	--	--	--	--	--	25	--	25

Students must select a course for Open Elective from a bucket provided by the Rahul Education Group.

Term Work Assessment: 25 Marks (Total marks) = 15 Marks (Case Study/Activity) + 5 Marks (Assignments) + 5 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
98431308	Entrepreneurship Development	2	-	-	30		-	30	60	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+IAT-2					
98431308	Entrepreneur-ship Development	--	--	--	--	--	25	--	25

Rationale :

Entrepreneurship is a crucial skill in the 21st century, fostering innovation, self-employment, and economic development. This course is designed to equip students with practical entrepreneurial competencies such as ideation, business model development, financial planning, legal frameworks, and pitching.

Course Objectives:

1. Understand and explain the fundamentals of entrepreneurship and startup ecosystems.
2. Generate, validate, and refine business ideas using structured techniques.

3. Develop comprehensive business plans and financial strategies.
4. Explore legal and funding options for startups in India.
5. Apply marketing, digital tools, and CRM for business growth.
6. Design prototypes and present business pitches to stakeholders.

Course Outcomes:

On successful completion of the course learner will be able to:

1) Explain entrepreneurial concepts and analyze types and traits of entrepreneurs.
2) Apply ideation techniques and validate business ideas using customer discovery tools.
3) Create business plans with financial projections and risk analysis.
4) Evaluate funding sources and legal requirements for a startup.
5) Implement digital marketing and branding strategies effectively.
6) Prepare and deliver compelling business pitches with basic prototyping.

Prerequisite: Fundamentals of communication and leadership skills.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Fundamentals of communication and leadership skills.		
I	**Introduction to Entrepreneurship**	Definition, Characteristics, and Types of Entrepreneurs. Entrepreneurial Motivation and Traits. Start-up Ecosystem in India. Challenges in Entrepreneurship	4	CO1
		Self-learning Topics: Evolution of Indian startup ecosystem post-2010 **Case Study**: Nirma – A middle-class entrepreneur revolutionizing detergents.	4	

II	**Business Idea Generation & Validation**	Ideation Techniques: Design Thinking, Brainstorming, Mind Mapping. Business Model Canvas (BMC). Market Research & Customer Validation. Minimum Viable Product (MVP) Concept.	5	CO2
		Self-learning Topics: Lean Canvas, Persona Creation **Case Study:** Airbnb – Validating the concept with MVP	5	
III	**Business Planning & Strategy**	Writing a Business Plan. SWOT Analysis and Competitive Analysis. Financial Planning and Budgeting. Risk Assessment and Management.	5	CO3
		Self-learning Topics: Financial forecasting using Excel/Google Sheets. **Case Study:** Zomato – Expansion through mergers & strategic pivots	5	
IV	**Funding and Legal Framework**	Sources of Funding: Bootstrapping, Angel Investors, Venture Capital Government Schemes & Start-up IndiaInitiatives. Business Registration & Legal Formalities. Intellectual Property Rights (IPR) & Patents	6	CO4
		Self-learning Topics: DPIIT recognition process and types of funding **Case Study:Paytm –** From Wallet to IPO: Funding journey	6	

V	**Marketing & Digital Presence**	Branding and Digital Marketing. Social Media Marketing & SEO. Customer Relationship Management (CRM). E-commerce & Online Business Models.	5	CO5
		Self-learning Topics: Use of Canva, Google Analytics **Case Study:** Lenskart – Omnichannel e-commerce marketing	5	
VI	**Business Pitching & Prototype Development**	Pitch Deck Preparation & Presentation Techniques. Prototyping with Open-source Tools. Elevator Pitch & Investor Pitch.	5	CO6
		Self-learning Topics: Canva for pitch deck or Successful Start-ups **Case Study:** Dunzo – MVP to final app evolution	5	
Total			30	

Text Books:

1. "Entrepreneurship Development and Small Business Enterprises" – Poornima M. Charantimath, Pearson, 3rd Edition, 2021.
2. "Innovation and Entrepreneurship" – Peter F. Drucker, Harper Business, Reprint Edition, 2019.
3. "Startup and Entrepreneurship: A Practical Guide" – Rajeev Roy, Oxford University Press, 2022.
4. "Essentials of Entrepreneurship and Small Business Management" – Norman Scarborough, Pearson, 9th Edition, 2021.
5. "The Lean Startup" – Eric Ries, Crown Publishing, 2018.

References:

1. "Disciplined Entrepreneurship: 24 Steps to a Successful Startup" – Bill Aulet, MIT Press, 2017.
2. "Zero to One: Notes on Startups, or How to Build the Future" – Peter Thiel, 2014.
3. "The $100 Startup" – Chris Guillebeau, Crown Business, 2019.
4. "Business Model Generation" – Alexander Osterwalder & Yves Pigneur, Wiley, 2020.
5. "Blue Ocean Strategy" – W. Chan Kim & Renée Mauborgne, Harvard Business Review Press, 2019.

Online References:

Sr. No.	Website Name
1.	Startup India Portal – https://www.startupindia.gov.in
2.	MIT OpenCourseWare – Entrepreneurship – https://ocw.mit.edu/courses/sloan- school-of-management/
3.	Coursera – Entrepreneurship Specialization – https://www.coursera.org/specializations/entrepreneurship

Assessment:

Term Work: Term Work shall consist of at least 4 to 5 **Case Study Analysis, Presentation** based on the above list. Also, Term work Journal must include at least 2 to 3 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Case Study report) + 5 Marks (Assignments) + 5 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned					Total Credits (C)
		L	T	P	L	T	P	SL	Notional Learning Hour	(Notional Learning Hour/30
98451309	**Environmental Science**	**2**	**-**	**-**	**30**		**-**	**30**	**60**	**2**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT - 1	IAT -2	IAT-1+IAT -2					
98451309	**Environmental Science**	20	20	40	60	2.5	--	--	100

Rationale :

Environmental Science provides a multidisciplinary perspective to understand, analyze, and solve ecological challenges. With growing global concern over climate change, pollution, and sustainable development, it is essential to equip students with the knowledge, skills, and values needed to contribute to sustainable solutions.

Course Objectives:

1. To understand the interrelationship between human activities and the environment.
2. To explore natural ecosystems and biodiversity conservation techniques.

3. To identify causes, effects, and mitigation strategies for various types of environmental pollution.
4. To analyze key environmental policies, laws, and sustainability practices.
5. To develop awareness, ethics, and responsibility towards environmental protection.
6. To promote sustainable development through informed decision-making and community participation.

Course Outcomes:

On successful completion of the course learner will be able to:

1) Explain core concepts of environmental science and ecological balance.
2) Identify sources and impacts of pollution on ecosystems and human health
3) Evaluate biodiversity importance and conservation strategies.
4) Analyze climate change, global warming, and their mitigation approaches.
5) Interpret government policies and sustainable practices for environment management.
6) Propose solutions for environmental issues at local and global levels.

Prerequisite: Basic Science Knowledge, General Awareness – About environmental issues, public policies, and civic responsibilities.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basic Science Knowledge, General Awareness – About environmental issues, public policies, and civic responsibilities.		
I	**The Multidisciplinary Nature of Environmental**	Definition, scope and importance. Need for public awareness. Role of information technology in environment and human health. Human population and the	5	CO1

	Studies	Environment. Population growth, variation among nations. Population Explosion- family welfare program. Environment and human health Women and child welfare.		
		Self-learning Topics: Environmental ethics **Case Study**: Chipko Movement (India) – Grassroot conservation.	5	
II	**Natural Resources**	Renewable and non-renewable resources. Natural resources & associated problems: a. Forest resources: b. Water resources: Natural resources & associated problems c. Mineral resources: d. Food resources: e. Energy resources: Role of an c. Mineral resources: d. Food resources: e. Energy resources: Role of an individual in conservation of natural resources: f. Equitable use of resources for sustainable lifestyles.	5	CO2
		Self-learning Topics: Carbon footprint calculators **Case Study:** Rainwater Harvesting	5	
III	**Ecosystems**	Concepts of an ecosystem. Introduction, types, characteristic features, structure and function of	5	CO3

		the following ecosystem: a. Forest ecosystem b. Grassland ecosystem c. Desert ecosystem d. Aquatic ecosystem (ponds, streams, lakes, rivers, Oceans, estuaries). Case study on various ecosystems in India.		
		Self-learning Topics: Biodiversity in Indian context **Case Study:** various ecosystems in India.	5	
IV	**Biodiversity and its Conservation**	Introduction-Definition: genetic species and ecosystem diversity. Bio-geographical classification of India Value of biodiversity : Consumptive use, productive use, social, ethical, aesthetic and option values, Bio-diversity at global, national, local levels India as a mega diversity nation Case study on Bio diversity in India.	5	CO4
		Self-learning Topics: Indoor pollution sources	5	
V	**Environmental Pollution Definition**	Causes of Pollution , effects and control measures of: a. Air pollution b. Water pollution c. Soil pollution. Solid waste management: Causes, effect and control measures of urban and industrial wastes. Role of an individual in prevention of pollution.	5	CO5

		Self-learning Topics: Greenhouse gases **Case study:** Pollution Disaster management: floods, earthquake, cyclone and landslides. Carbon Credits for pollution prevention	5	
VI	**Social Issues and Environment**	From unsustainable to sustainable development Urban problems related to energy, Water conservation, rain water harvesting, watershed management. Environmental ethics: issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Consumerism and waste products. Environment protection act. Public awareness.	5	CO6
		Self-learning Topics: Environmental audit methods. **Case Study:** Environmental Ethics.	5	
Total			**30**	

Text Books:

1. Environmental Science: Towards a Sustainable Future, G. Tyler Miller and Scott Spoolman, 13th Edition, Cengage Learning 2021
2. Environmental Management: Text and Cases, Bala Krishnamoorthy, 3rd Edition, PHI Learning, Publication Year: 2016
3. Green IT: Concepts, Technologies, and Best Practices, Markus Allemann 1. Springer 2008

4. Sustainable IT: Slimming Down and Greening Up Your IT Infrastructure, David F. Linthicum, IBM Press 2009
5. Environmental Modelling: Finding Solutions to Environmental Problems, David L. Murray, Cambridge University Press 2016
6. Remote Sensing and Image Interpretation, Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman, 9th Edition, John Wiley & Sons 2020
7. Business Ethics: Concepts and Cases, Manuel Velasquez, Pearson 2012

References:

1. Environmental Law and Policy in India, Shyam Divan and Armin Rosencranz, 2nd Edition, Oxford University Press 2018.
2. The International Handbook of Environmental Laws, David Freestone and Jonathon L. Rubin, Edward Elgar Publishing 2000.
3. E-Waste Management: Challenges and Opportunities in Developing Countries, Ruediger Kuehr and Ram K. Jain, Springer 2008.
4. The E-Waste Handbook: Managing Electronic Waste, Klaus Hieronymi, Ruediger Kuehr, and Ram K. Jain, Earth 2009.
5. Environmental Ethics: An Introduction, J. Baird Callicott, University of Georgia Press, 1999.

Online References:

Sr. No.	**Website Name**
1.	Centre for Science and Environment (CSE), Website: cseindia.org
2.	Ministry of Environment, Forest and Climate Change (MoEFCC), Government of India
3.	CSIR-National Environmental Engineering Research Institute (NEERI)

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30)
12812310	**IDEA LAB - 3(Innovation Design Engineering and Apply)**	1	--	2*	15	--	30	15	60	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-1	IAT-2	IAT-1+IAT-2				
12812310	**IDEA LAB - 3(Innovation Design Engineering and Apply)**	--	--	--	--	50	50	100

Rationale :

Aligned with the National Education Policy (NEP) 2020, the institution emphasizes experiential, interdisciplinary, and project-based learning through the IDEA Lab—a central hub for hands-on innovation.

To strengthen the undergraduate research ecosystem, the institution has adopted a theme-based academic model aligned with UN SGD. Each semester features six curated problem statements based on local need and aligned with core subjects in the same semester, enabling students to apply classroom knowledge to real-world challenges. Every student selects one problem and develops an individual, subject-integrated solution—enhancing both academic understanding and research skills.

The IDEA Lab supports this initiative with facilities for design thinking, prototyping, and product development. Students maintain a project logbook throughout the semester to track their progress and reflections.

To ensure academic accountability, a two-tier assessment framework is implemented:

- Project Assessment based on standardized IDEA Lab rubrics.
- Subject-Based Term Work Assessment focused on the application of same-semester subject knowledge in the project.

Lab Objectives:

1. To promote experiential and project-based learning that bridges theoretical knowledge with real-world problem-solving.
2. To encourage interdisciplinary integration by enabling students to apply concepts from multiple subjects within a single cohesive project.
3. To develop innovation and design thinking skills through hands-on activities and iterative solution development.
4. To foster critical thinking and creativity by engaging students in open-ended problems with multiple solution pathways.

5. To enhance communication, collaboration, and documentation skills essential for professional engineering practice.
6. To build an entrepreneurial and research mindset by guiding students to develop scalable, socially-relevant, and technically viable prototype

Lab Outcomes: Student will be able to

1. Recall and articulate key concepts from core and allied subjects relevant to the assigned project.
2. Explain the interdisciplinary nature of the problem and the role of each subject in addressing it.
3. Apply appropriate tools, techniques, and theoretical knowledge to develop project components.
4. Analyze problem constraints and user requirements to structure a feasible and efficient solution.
5. Evaluate multiple design options and justify the chosen solution based on technical and practical considerations.
6. Create a functional prototype or solution that demonstrates innovation, utility, and integration of interdisciplinary knowledge

1) Guidelines for IDEA Project

a) Project Guidelines (Interdisciplinary Project Execution in IDEA Lab)

- Each student works on an individual interdisciplinary project aligned with the semester theme.
- Faculty in-charges for the IDEA Lab are assigned according to the complexity of the project and the capacity of the respective departments.
- Faculty in-charges mentor both the academic and technical aspects, and track weekly progress.
- Project assessment will be rubric-based, ensuring depth, innovation, documentation, and

ownership.

- Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
- Faculty in-charges must attend relevant FDPs to ensure uniformity in mentoring and evaluation.

b) Guidelines for same semester Subject Concepts Applied within the Project

- Termwork for each subject will partially reflect how well a student applies subject-specific concepts in their interdisciplinary project.
- Internal assessment panel will collaborate to align project components with subject learning outcomes.

---

c) Role of Faculty In-Charges in IDEA Lab Projects

Faculty in-charges play a pivotal role in the success of interdisciplinary, theme-based projects under the IDEA Lab. Their responsibilities extend beyond technical supervision to include academic alignment, innovation facilitation, and active student engagement. Their key roles include:

1. **Motivating and Inspiring Students**
    - Encourage students to take ownership of their learning and projects.
    - Cultivate a mindset of curiosity, exploration, and social relevance.
    - Foster an environment where students feel empowered to take creative risks.
2. **Conducting Brainstorming and Ideation Sessions**
    - Organize structured brainstorming sessions at the start of the semester to help students define their problem statements and solution pathways.
    - Promote collaborative thinking, design exploration, and interdisciplinary integration.
3. **Arranging Guest Lectures and Expert Talks**
    - Identify and invite industry experts, researchers, and innovators for guest lectures aligned with the semester's theme or subject areas.
    - Facilitate exposure to real-world challenges, current trends, and future opportunities.

4. **Ensuring Uniqueness and Originality of Projects**
   - Actively review proposed ideas to ensure **no duplication of solutions** across students.
   - Encourage students to explore novel approaches, technologies, and perspectives.
5. **Promoting Discussion and Collaborative Learning**
   - Create platforms for students to present, discuss, and receive peer and mentor feedback.
   - Facilitate idea refinement through regular discussions and group engagement.
6. **Aligning Subject Content Beyond Syllabus**
   - Faculty in-charges must **align subject content beyond the syllabus of the same semester** with the **IDEA Lab theme and assigned problem statements**.
   - This ensures relevance, depth, and meaningful interdisciplinary integration.
7. **Same Semester Faculty Requirement**
   - Faculty in-charges must be teaching subjects in the **same semester** as the students' project to ensure seamless academic integration and contextual understanding.
8. **Monitoring and Documentation**
   - Oversee project logbook maintenance, milestone tracking, and submission of progress reports.
   - Provide ongoing feedback and ensure project alignment with learning outcomes.
9. **Coordination with Subject Faculty**
   - Work in collaboration with other subject faculty to help students embed theoretical and practical aspects of their coursework into the project.
   - Facilitate subject-term mapping and contribute to termwork assessment based on evidence.

2) Implementation Strategy

a) Project Implementation in IDEA Lab

Aspect	Implementation Strategy
Faculty in-charges	**Faculty in-charges assigned based on project nature and department capacity.**
Mentoring Role	**Faculty in-charges oversee academic/technical development, interdisciplinary integration, and timely documentation.**

Aspect	Implementation Strategy
Capacity Building	**Faculty in-charges undergo workshops on design thinking, innovation, assessment rubrics, and outcome-based mentoring.**
Assessment Contribution	**Faculty in-charges contribute to 25 marks allocated for the IDEA Lab project termwork. The remaining assessments are conducted by the external examiner.**
Recognition & Incentives	**Faculty in-charges receive workload credits or are formally acknowledged in performance reviews.**

---

b) Implementation of Subject-Term Work Mapping within Projects

Aspect	Implementation Strategy
Mapping Subject Outcomes	**Faculty in-charges align their content beyond syllabus with the student's project by coordinating with the assigned project guide.**
Independent Evaluation	**Internal assessment panel evaluate students based on their application of subject-specific concepts within the project. This contributes to a** separate 25 marks allocated for termwork based on subject **application.**
Evidence Sources	**Evaluation is supported by project logbooks, subject-specific deliverables (e.g., tools, simulations, models), and review presentation inputs.**
Outcome Assurance	**Ensures practical demonstration of subject understanding and its integration into the interdisciplinary solution.**

Implementation Notes:

- Guide faculty assess their course's contribution using specific evidence such as:
  - Logbooks
  - Subject-specific outputs (e.g., simulations, designs)

- Paper publications or review presentations

2) Guidelines for Assessment

Two-tier rubrics are applied independently to evaluate subject concept application and innovation within the project.

a) Assessment of IDEA Lab Projects (Individual Interdisciplinary Projects) (25 Marks)

Presentation-Based Assessment Structure (Total: 25 Marks)

Assessment Month	Weightage	Marks
Month 1 (Formative 1)	20%	5 marks
Month 2 (Formative 2)	40%	10 marks
Month 3 (Formative 3)	40%	10 marks

Rubric-Based Evaluation Criteria

Criteria	Month 1 (5)	Month 2 (10)	Month 3 (10)
Problem Understanding	Connects problem to subjects	Defines interdisciplinary scope	Demonstrates deep conceptual grasp
Subject Knowledge Application	Identifies relevant concepts	Applies principles in design	Integrates multiple subject areas correctly
Innovation & Design Thinking	Proposes creative idea	Develops and tests feasible solution	Final solution shows originality and utility
Documentation & Presentation	Logbook initiated, plan presented	Mid-design log and visuals	Final report and demo completed
Progress & Ownership	Meets deadlines, shows planning	Demonstrates self-motivation	Completes project independently with reflection

b) Term Work Assessment of Subject Concepts Applied in Projects (25 Marks)

Applicable to All Subjects Integrated with Interdisciplinary Projects

To reflect meaningful application of subject knowledge, each subject will be assessed through the following rubric:

Criteria	**Marks**	**Description**
Subject Knowledge Application	8	Depth and accuracy of concept integration into the project
Practical Design or Tool Usage	5	Use of subject-specific hardware/software/simulation/tools
Documentation	4	Quality and clarity of subject-related logs and reports
Viva/Presentation	4	Ability to explain subject's relevance and role in the project
Continuous Engagement	4	Evidence of consistent participation via logbooks and feedback

c) Total Assessment Structure

Component	Marks	Assessed By
Termwork – Project Execution	25 Marks	Project Guide
Termwork – Application of Subject Concepts	25 Marks	IDEA Lab Panel
Viva Voce (Final Evaluation)	50 Marks	External Examiner

Semester-IV

Theme: CrisisNet: Rapid Disaster Response & Aid Coordination

Aligned with UNSDG : SDG 11 (Sustainable Cities and Communities), SDG 13 (Climate Action), and SDG 16 (Peace, Justice, and Strong Institutions)

Keywords: Natural disasters, emergency response teams, resource allocation, rescue, Relief materials, Disaster warnings.

The theme "CrisisNet: Rapid Disaster Response & Aid Coordination" is a critical framework for leveraging technology and organized efforts to mitigate the devastating impacts of various crises, from natural disasters to other large-scale emergencies. It's about creating a robust, interconnected system that ensures swift, efficient, and equitable support to affected populations, thereby strengthening community resilience and stability.

It aligns profoundly with UN Sustainable Development Goals (SDGs) SDG 11 (Sustainable Cities and Communities), SDG 13 (Climate Action), and SDG 16 (Peace, Justice, and Strong Institutions), particularly in the context of building resilient and peaceful communities.

Alignment with UNSDG 11: Sustainable Cities and Communities

SDG 11 aims to "make cities and human settlements inclusive, safe, resilient and sustainable." "CrisisNet" directly contributes to this by:

- **Enhancing Urban and Rural Resilience:** Disasters (both natural and man-made) can cripple communities. CrisisNet's focus on rapid response and aid coordination means that cities and communities, whether urban centers like Mumbai or remote rural villages in Maharashtra, can better withstand and recover from shocks. By facilitating quick assessment of damage, efficient deployment of resources, and effective communication, it reduces the long-term impact of disasters on infrastructure, livelihoods, and social fabric.
- **Protecting Vulnerable Populations:** Disasters disproportionately affect the most vulnerable segments of society. CrisisNet, through its efficient coordination mechanisms, ensures that aid reaches those who need it most, including the poor, elderly, disabled, and women and children, thereby contributing to inclusive settlements where no one is left behind in times of crisis.
- **Improving Disaster Risk Reduction and Management:** A rapid and coordinated response system is a cornerstone of effective disaster risk reduction. CrisisNet can integrate early warning systems, facilitate the dissemination of crucial information, and support the evacuation and sheltering of affected populations, thereby minimizing casualties and property damage and making communities safer.

- **Strengthening Community Infrastructure and Services:** Post-disaster recovery relies on effective aid. By streamlining aid coordination, CrisisNet helps in the swift restoration of essential services (water, sanitation, electricity, healthcare) and the rebuilding of resilient infrastructure, ensuring that communities can bounce back quickly and sustainably.

Alignment with UNSDG 13: Climate Action

SDG 13 calls for "taking urgent action to combat climate change and its impacts." "CrisisNet" plays a crucial role here because:

- **Addressing Climate Change-Induced Disasters:** Climate change is increasing the frequency and intensity of extreme weather events such as floods, droughts, heatwaves, and cyclones, which are common in regions like Maharashtra. CrisisNet provides the necessary framework for responding effectively to these climate-induced disasters, mitigating their immediate human and economic toll.
- **Facilitating Adaptation and Resilience Building:** While primarily a response mechanism, the data and insights gathered through CrisisNet's operations can inform long-term climate adaptation strategies. Understanding patterns of damage, aid requirements, and community vulnerabilities helps in planning resilient infrastructure and developing early warning systems tailored to climate-related risks, thereby enhancing overall adaptive capacity.
- **Minimizing Loss and Damage:** Rapid and effective aid coordination reduces the "loss and damage" caused by climate-related events. By ensuring timely medical attention, food, shelter, and psychological support, CrisisNet directly lessens the human suffering and economic disruption that climate change inflicts on communities.
- **Promoting Data-Driven Climate Response:** CrisisNet, as a technology-driven system, can collect real-time data on disaster impacts. This data is invaluable for climate scientists and policymakers to understand the localized effects of climate change, refine climate models, and develop more targeted and effective climate action plans.

Alignment with UNSDG 16: Peace, Justice, and Strong Institutions

SDG 16 aims to "promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels." "CrisisNet" contributes significantly by:

- **Building Effective and Accountable Institutions:** CrisisNet is fundamentally about creating a more organized, transparent, and efficient system for disaster response. This requires strong institutional frameworks, clear lines of authority, and accountability

mechanisms for aid distribution and resource management, directly contributing to more effective and trustworthy governance during crises.

- **Enhancing Transparency and Reducing Corruption in Aid:** Disasters can sometimes create environments ripe for mismanagement or corruption. CrisisNet, through digital tracking, data transparency, and coordinated efforts, can significantly reduce opportunities for aid diversion, ensuring that resources reach their intended beneficiaries and fostering trust in the institutions managing the response.
- **Strengthening Partnerships and Collaboration:** Effective disaster response necessitates collaboration between government agencies, NGOs, civil society organizations, and local communities. CrisisNet acts as a platform for fostering these partnerships, improving communication, and ensuring that all stakeholders work cohesively towards common goals, leading to more inclusive and coordinated responses.
- **Ensuring Access to Information and Justice:** In the aftermath of a disaster, access to accurate information is crucial for survival and recovery. CrisisNet can provide a reliable channel for disseminating vital alerts, instructions, and information about aid availability. Furthermore, by ensuring proper aid distribution and addressing grievances related to the response, it indirectly supports the principle of justice for affected populations.
- **Promoting Peaceful Recovery:** Disasters can exacerbate existing social tensions or create new ones due to resource scarcity or perceived inequalities in aid distribution. A well-coordinated and transparent system like CrisisNet helps in equitably distributing resources and managing expectations, thus contributing to more peaceful and cohesive recovery processes.

Problem Statements:

1. **During natural disasters, emergency response teams (police, fire, ambulances) often lack a unified command center, leading to duplicated efforts and missed rescue operations.**
2. **Lack of updated satellite or drone imagery prevents accurate assessment of disaster-hit zones, delaying resource allocation and rescue.**
3. **Relief materials (food, water, medicine) are either oversupplied in some areas or don't reach the actual victims in time.**
4. **Authorities release updates through different channels (Twitter, SMS, radio), but inconsistent messaging creates confusion among citizens.**
5. **Disaster warnings issued only in English are not understood by a significant portion of the population, delaying their ability to comprehend the threat and respond effectively.**
6. **Highway accidents often result in severe injuries where every minute counts. Ambulances frequently arrive late resulting in preventable fatalities.**
   Rumors of exaggerated dangers spread quickly during crises causing unnecessary panic and misdirection of scarce resources.

Post-disaster analysis reports are often buried in static documents, preventing planners from systematically using past insights to improve future preparedness.

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per week)			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12211401	Sampling Theory and Optimization	2	1	--	30	15	--	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Semester Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+IAT-2					
12211401	Sampling Theory and Optimization	20	20	40	60	2.5	25	--	125

Rationale:

In a second-year Computer Engineering course, understanding statistical tests and optimization techniques is fundamental for analyzing system performance and making informed design decisions. Large and small sample tests enable students to evaluate hypotheses about system behavior based on the size of data samples, such as testing network latency or processor reliability. The Chi-square and F-tests provide tools to examine categorical data relationships and compare variances, which are useful in validating error distributions and algorithm performance consistency. Optimization methods like the Simplex method and Duality assist in solving linear programming problems, such as optimizing CPU time allocation or resource management, while Nonlinear Kuhn-Tucker conditions extend these techniques to complex systems with nonlinear constraints, like balancing power and speed in embedded devices. Markov Chains offer a way to model stochastic processes where the next state depends only on the current state, helping to analyze network traffic and system states. Finally, Queuing Theory is essential for studying

waiting lines in processors or network buffers, enabling engineers to optimize task scheduling and improve system throughput. Together, these concepts equip computer engineering students with practical tools to design, analyze, and optimize computing systems effectively.

Course Objectives:

1. To explain the principles of hypothesis testing for large and small samples to analyze computer system data.
2. To describe the use of Chi-square and F-tests for analyzing variance and independence in datasets.
3. To understand linear programming concepts to optimize resource allocation in computing tasks.
4. To explain the Kuhn-Tucker conditions to handle constraints in non-linear optimization problems.
5. To understand describe Markov chain models to represent stochastic processes in computing.
6. To understand queuing models to evaluate performance in computing environments.

Course Outcomes:

1. Students will be able to apply large and small sample tests to evaluate performance metrics like CPU utilization and network latency.
2. Students will be able to perform chi-square tests for validating error distributions in software and conduct f-tests to compare memory usage variances under different workloads.
3. Students will be able to solve optimization problems like scheduling or bandwidth allocation using the simplex method and interpret dual problems for constraint analysis.
4. Students will be able to formulate and analyze constrained non-linear optimization problems such as minimizing power consumption in embedded systems.
5. Students will be able to model system state transitions such as cache hits/misses or network routing behaviors using Markov chains.
6. Students will be able to analyze and optimize the performance of servers and network routers through queuing theory examples like M/M/1 and M/M/c queues.

Prerequisite:

1. Basic Probability Theory, Matrix Algebra.

R-2025- S.E Computer Engineering

DETAILED SYLLABUS

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	Sampling theory-I	Sampling distribution, Test of Hypothesis, Level of Significance, Critical region, One-tailed, and two tailed test, Degree of freedom.	05	CO1
		Test significance for large samples. Test the significance of mean.		
		Students' t-distribution (Small sample). Test the significance of single sample mean and two independent sample means and paired t- test)		
		Self-learning Topics: 1. Estimate parameters of a population. 2. Test of significance of large samples, Proportion test. 3. Difference between the means of two large samples.	10	
II	Sampling theory-II	Chi-square test: Test of goodness of fit and independence of attributes (Contingency table) including Yate's Correction.	05	CO2
		Analysis of variance: F-test (significant difference between variances of two samples).		
		Self-learning Topics: 1. ANOVA**:** One way classification, 2. ANOVA**:** Two-way classification (short-cut method).	10	
III	Linear Programming Problems	Types of solutions, Standard and Canonical of LPP, Basic and Feasiblesolutions, slack variables, surplus variables.	05	CO3
		Simplex method.		

		Duality, Dual of LPP.		
		Dual Simplex Method.		
		Self-learning Topics: 1. Artificial variable method, Big M method. 2. Sensitivity Analysis, Two-Phase Simplex Method, Revised Simplex Method.	10	
IV	Nonlinear Programming Problems	NLPP with one equality constraint (two or three variables) using the method of Lagrange's multipliers	05	CO4
		NLPP with one inequality constraint (two variables): Kuhn-Tucker conditions.		
V	Markov Chains	Introduction to Markov Chains	05	CO5
		Transition Probability matrix, Transition Diagram		
		Homogeneous Markov Chains		
		Chapman-Kolmogorov Equations for discrete Markov chains.		
		Self-learning Topics: 1. Initial Conditions 2. Classification of states 3. Semi-Markov Chains 4. Markov Processes State Space and Continuous time.	10	
VI	Queuing Theory	Introduction to Queuing Systems	05	CO6
		Birth-Death Processes & M/M/1 Queues		
		M/M/c and Finite Queue Models		
		M/G/1 and G/G/1 Queues		

		Queuing Networks		
		Self-learning Topics: 1. Priority Queues. 2. Queueing Disciplines. 3. Retrial Queues & Balking/Reneging 4. Queuing with Server Vacations	10	

Text Books:

1. Operations Research, Hira and Gupta, S. Chand Publication.

2. Fundamentals of Statistics, S. C. Gupta.

3. Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa.

4. Murray Spiegel, “Schaum's Outline of Probability and Statistics”, 4th Edition, Tata McGraw-Hill.

References:

1. Advanced Engineering Mathematics, Erwin Kreyszig, John Wiley & Sons.

2. Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill education.

3. Operations Research: An Introduction, Hamdy A Taha, Pearson.

4. Engineering Optimization: Theory and Practice, S.S Rao, Wiley-Blackwell.

5. S. D. Sharma, “Operations Research”, Kedarnath Ramnath Publishing, Meerut, 8th Edition, 2015.

Online References:

Sr. No.	Website Name
1.	https://archive.nptel.ac.in/courses/117/103/117103017/
2.	https://archive.nptel.ac.in/courses/111/102/111102111/

General Instructions:

1. Batch wise tutorials are to be conducted. The number of students per batch should be as per the university pattern for practical.
2. Students must be encouraged to write **at least 10** class tutorials on entire syllabus.
3. A group of 4-6 students should be assigned a self-learning topic. Students should prepare a presentation/problem solving of 10-15 minutes. This should be considered as mini project.

This project should be graded for 10 marks depending on the performance of the students.

The distribution of Term Work marks will be as follows –

1. Regularity and active involvement (Theory and Tutorial) 05 marks.
2. Class Tutorials on entire syllabus 10 marks.
3. Mini Project 10 marks.

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12211402	**Software Engineering & Project Management**	3	--	--	45	--	--	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+IAT-2					
12211402	**Software Engineering & Project Management**	20	20	40	60	2.5	--	--	100

Rationale :

This course equips students with essential software engineering and project management skills. It strategically focuses on Sustainable Digital Supply Chains and Smart Logistics, addressing a

critical industry demand. The curriculum aligns with NEP 2020 by offering holistic, multidisciplinary education. It fosters skill-based learning, critical thinking, and problem-solving through practical applications. Emphasis on experiential learning is achieved via real-world case studies and a dedicated project module. Crucially, the course directly contributes to UN Sustainable Development Goals like Responsible Consumption (SDG 12) and Sustainable Cities (SDG 11). It also promotes Climate Action (SDG 13) and Innovation (SDG 9) within logistics. The pedagogical approach, guided by Bloom's Taxonomy, ensures comprehensive cognitive development. Graduates will be competent, responsible, and future-ready professionals for a sustainable global economy.

Course Objective:

1. Comprehend the foundational principles of software development, including various software life cycle process models, agile methodologies, SCRUM, and other contemporary agile practices.
2. Apply methods for eliciting, specifying, visualizing, and analyzing software requirements, ensuring comprehensive understanding of stakeholder needs.
3. Design and implement software solutions utilizing core concepts and principles of software design, emphasizing user-centric approaches and principles of effective user interfaces.
4. Evaluate and select appropriate software testing strategies and tactics, implement quality assurance processes, and manage software configurations effectively.
5. Articulate the critical role of project management within the software development life cycle (SDLC) and assess the significance of Software Engineering in the overall Product Life Cycle (PLC).
6. Develop and optimize project schedules, construct network diagrams (AON, AOA), utilize CPM and PERT techniques, and mitigate project risks by organizing activities based on risk impact factors

Course Outcomes:

1. Categorize various software application domains and differentiate between diverse process models employed in software development (L2, L4)

2. Formulate precise software specifications, classify distinct types of software requirements, and employ effective requirement gathering techniques. (L3, L4)
3. Translate requirements models into robust design models and demonstrate the application of sound software and user-interface design principles. (L3, L6)
4. Differentiate between Software Configuration Management (SCM) and Software Quality Assurance (SQA), evaluate various testing strategies and tactics, and compare their effectiveness. (L4, L5)
5. Justify the integral role of SDLC in Software Project Development and evaluate the paramount importance of Software Engineering within the overall Product Life Cycle.(L5)
6. Construct and optimize project schedules, design and develop network diagrams for diverse project types, and prioritize different project activities based on their risk impact factor. (L3, L5, L6)

Prerequisite:

DETAILED SYLLABUS:

R-2025- S.E Computer Engineering

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Nature of Software, Software Definition, Software Characteristics, Software Application Domains	-	-
I	Introduction to Software Engineering	Evolution and scope of software engineering. Software process models: Waterfall, Spiral, V-Model, Agile, DevOps. Characteristics of good software (reliability, maintainability, efficiency, etc.) Software engineering ethics and professional conduct. Overview of SDLC and software process activities.	8	CO1
		Self-learning Topics: History and Milestones of Software Engineering. In-depth Study of Agile & DevOps Practices. DevOps for Agile Software Delivery. Explore the principles and practices of DevOps, and how they enhance agile methodologies (Scrum, XP) for continuous integration, delivery, and deployment in modern supply chain software	6	
II	Requirements Engineering and Feasibility Analysis	Functional vs Non-functional requirements. Elicitation techniques: Interviews, questionnaires, use cases. Writing Software Requirements Specification (SRS) using IEEE standards. Feasibility analysis: Technical, economic, operational. Requirements validation and change management.	7	CO1, CO2

		Self-learning Topics: AI-Powered Requirements Analysis and Automation: Investigate tools and techniques that use AI/ML to assist in requirements gathering (e.g., sentiment analysis of user feedback) and automated test case generation for complex logistics systems.	6	
III	Software Design and Architecture with Implementation	Design principles: Modularity, cohesion, coupling, abstraction. UML diagrams: Class, Use Case, Sequence, Activity. Design patterns: Singleton, Factory, MVC. Software architecture styles: Layered, Client-Server, Microservices. Formulate Architecture Design, Component Level Design, System Level Design. Create User Interface Designs focusing on usability and accessibility. Coding standards (PEP8, MISRA), naming conventions, comments. Procedural vs Object-Oriented Programming (OOP) design.	8	CO1, CO2, CO3, CO4
		Self-learning Topics: Compare Procedural vs. Object-Oriented Programming. Hands-on with Code Reusability Techniques. Study Microservices Architecture in Large-Scale Logistics Systems: Research the benefits and challenges of designing large-scale, distributed logistics applications	8	
IV	Software Testing & Quality Assurance	Software Testing Life Cycle (STLC). Testing types: Unit, Integration, System, Acceptance. White-box & Black-box techniques. Evaluating quality metrics like defect density and test coverage and testing tools such as JUnit (Java), PyTest (Python), and Selenium for automation. Examine the role	8	CO5

		of reviews, inspections, and audits in ensuring software quality		
		Self-learning Topics: Practice Black-box and White-box Testing Techniques. Study Cybersecurity in IoT-Enabled Supply Chains	7	
V	Software Project Management & Maintenance	Introduction to software project management principles.Software project lifecycle: Initiation → Closure. Estimation techniques: COCOMO, Function Point. Gantt Charts, PERT, CPM for scheduling. Risk management and stakeholder communication strategies. Building a Project Charter and Work Breakdown Structure (WBS). Types of maintenance: Corrective, Adaptive, Preventive. Configuration and change management tools.	7	CO1, CO2, CO3, CO4, CO6
		Self-learning Topics: Create a project plan with WBS, estimation & risk register. Risk Management with Artificial Intelligence: Explore how AI and predictive analytics can be used to identify, assess, and mitigate project risks more effectively	6	

R-2025- S.E Computer Engineering

VI	Practical Application - Theme Based Case Studies Theme: Sustainable Digital Supply Chains & Smart Logistics Project	AI / ML Integration DevOps, CI/CD & Microservices Cloud-Native & IoT Systems Blockchain & Security in Software Engineering	7	CO1, CO2, CO3, CO4, CO5, CO6
		Case Study 1: Optimizing Last-Mile Delivery for Urban Sustainability Scenario: A large e-commerce company faces challenges with inefficient last-mile delivery in dense urban areas, leading to high fuel consumption, increased carbon emissions, traffic congestion, and customer dissatisfaction due to delayed deliveries. They decide to invest in a "Smart Logistics" software solution. Case Study 2: Implementing a Digital Twin for Green Manufacturing & Supply Chain Transparency Scenario: A manufacturing company producing electronic components aims to achieve greater sustainability by reducing waste, optimizing energy consumption, and ensuring ethical sourcing throughout its supply chain. They decide to develop and implement a "Digital Twin" of their manufacturing process and supply chain. Case Study 3: Blockchain for Ethical Sourcing and Supply Chain Traceability	12	

		Scenario: A fashion brand commits to ensuring ethical sourcing of raw materials (e.g., cotton, dyes) and transparent supply chain practices to prevent child labor and environmental damage. They decide to implement a blockchain-based traceability system for their supply chain. Case Study 4: AI-Powered Demand Forecasting for Waste Reduction in Food Supply Chain Scenario: A large supermarket chain struggles with significant food waste due to inaccurate demand forecasting, leading to overstocking and expired products. They aim to implement an AI/Machine Learning-driven demand forecasting system to optimize inventory and reduce waste. Case Study 5: IoT-Enabled Cold Chain Monitoring for Pharmaceutical Logistics Scenario: A global pharmaceutical company needs to ensure that sensitive vaccines and medicines are maintained within precise temperature ranges throughout their journey from manufacturing plants to hospitals and clinics worldwide. Any deviation can render Case Study 6: AI-Driven Warehouse Automation for Circular Economy Initiatives		

		Scenario: A company specializing in electronics recycling and refurbishment aims to increase its processing efficiency and material recovery rates. They want to implement an "AI-driven warehouse automation" system to sort incoming electronic waste, identify Case Study 7: Predictive Maintenance for Fleet Management in Green Logistics Scenario: A large logistics company with a fleet of delivery vehicles wants to reduce operational costs, minimize vehicle downtime, and lower carbon emissions by proactively identifying and addressing potential mechanical failures. They aim to implement a "Predictive Maintenance" system using IoT and AI.		

Text Books:

1. Roger S Pressman, "Software Engineering: A Practitioner's Approach," 7th Edition, McGraw-Hill. ISBN: 0073375977.
2. Jack T. Marchewka, "Information Technology Project Management," 4th Edition, Wiley India.
3. Ian Sommerville " Software Engineering" 9th edition Pearson Education SBN-13: 978-0-13-703515-1, ISBN-10: 0-13-703515-2

References:

1. Pankaj Jalote, "Software Engineering: A Precise Approach," Wiley India.
2. Ian Sommerville, "Software Engineering," 9th Edition, Pearson Education. ISBN-13: 978-0-13-703515-1, ISBN-10: 0-13-703515-2.

3. John M. Nicholas, "Project Management for Business and Technology," 3rd Edition, Pearson Education.
4. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management."

Online References:

Sr. No.	Website Name
1.	https://www.ireb.org/en/ (International Requirements Engineering Board (IREB) - CPRE Syllabus)
2.	https://martinfowler.com/ (Martin Fowler's website)
3.	https://www.pmi.org/ (Project Management Institute (PMI))

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits(C) (Notional Learning Hour/30
12211403	Microprocessor & Microcontroller	3	--	--	45	--	--	45	90	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+IAT-2					
12211403	Microprocessor & Microcontroller	20	20	40	60	2.5	--	--	100

Rationale :

Microprocessors form the backbone of computer engineering and embedded systems, enabling students to understand hardware-software integration. With applications in disaster response—such as early warning systems and emergency communication—this course, integrated with the **CrisisNet** theme, equips students to build rapid, resilient solutions for crisis management, supporting **SDG 11** and **SDG 13**.

Course Objectives:

1) Understand the evolution, architecture, and applications of microcomputer systems and Intel microprocessors**.**

2) Analyze the software architecture and memory organization of the 8086/88 microprocessors.

3) Apply knowledge of the instruction set and addressing modes to develop assembly language programs.

4) Design microprocessor-based systems using memory and I/O interfacing techniques.

5) Interface peripheral devices and controllers with microprocessors.

6) Understand the role and function of co-processors and advanced Intel microprocessors.

Course Outcomes:

Students will be able to-

1) Describe microprocessor architecture and its relevance in emergency systems.

2) Develop optimized assembly code for time-critical and modular disaster response tasks.

3) Design memory and peripheral interfaces for real-time monitoring and control.

4) Interface and program peripheral controllers to manage distributed sensors and communication.

5) Create mini-projects simulating components of CrisisNet (e.g., emergency alert devices, field routers).

6) Apply embedded techniques to build hardware prototypes for disaster-prone areas.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Digital Electronics, Basic Computer Architecture		
I		• General microcomputer	8	CO1

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
	Introduction to Microprocessors and Microcomputers	architecture ● Evolution of Intel microprocessors (up to 8086/88) ● Stored program concept ● Applications of microcomputers. Real-world examples: Drones, emergency buttons, field terminals		
		Self-learning topics- ● Timeline and comparison of Intel processor generations (8086 to modern x86) ● Case studies on microprocessor use in real-time crisis systems (e.g., earthquake detectors)	8	CO1
II	8086/88 Software Architecture	● Memory segmentation, types of memory (dedicated, reserved, general use) ● Data types and memory organization	8	CO2

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
		• Addressing modes for sensor data and controller commands • Instruction set design: emphasis on control transfer, string handling in signal processing		
		Self-learning topics- • Intel addressing mode animations/tutorials (Intel Developer docs) • Microprocessor behavior under low-power and emergency modes (e.g., sleep mode, reset handling)	8	CO2
III	Assembly Language Programming	• Programming process: Specification, design, coding, assembling, testing, debugging • Modularization of problems and use of procedures and macros • Arithmetic and logical programming	7	CO3

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
		• Delay generation, use of data tables • Programming examples and case studies- Fire sensor polling, Alert broadcasting loop, Portable GPS data encoder		
		Self-learning topics- • Use of procedures and macros for sensor data parsing and alert message formatting • Exploration of open-source disaster alert code snippets in x86 assembly	7	CO3
IV	8086/88 Hardware Architecture	• Three-bus architecture • Clock and reset generation • Bus types and buffering techniques • Minimum and maximum mode operations. Design considerations for disaster-	8	CO4

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
		prone environments		
		Self-learning topics- ● PCB design basics and layout tools for microprocessor boards (e.g., KiCAD) ● System clock synchronization in distributed sensor systems	8	CO4
V	Memory and I/O Interfacing	● Interfacing techniques: Memory and I/O module design ● Parallel and serial I/O ● Programmed I/O, interrupt-driven I/O (using 8279) ● DMA-based data transfer (using 8237)	7	CO5
		Self-learning topics- ● Animation-based learning of interrupt cycle vs polling ● Serial vs parallel communication – pros & cons	7	CO5

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
		in crisis response systems		
VI	Peripheral Controllers and Coprocessors	Programming and interfacing of: ● 8237 (DMA for alerts) ● 8251 (Serial communication for data modems) ● 8255 (Sensor interface for water/flood levels) ● 8259 (Interrupt management for event-driven response) ● 8279 (Keypad/display in field nodes) ● Introduction to 8087 (numeric processing for terrain data) ● 8089 (I/O processor for aid distribution stations) ● Conceptual overview of integrating 80186/80286 in scalable networks	7	CO6
		Self-learning topics-	7	CO6

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
		• Real-world interfacing of water level sensors using 8255/ADC • Comparison of serial protocols (RS232, I2C, SPI) for field devices		

Text Books:

1. The 8086/8088 Family: Design, Programming and Interfacing, John Uffenbeck, PHI Learning, 2nd Edition, 2002

2. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications, Walter A. Triebel and Avtar Singh, PHI Learning, 4th Edition, 2002

3. Microprocessor and Interfacing: Programming and Hardware, Douglas V. Hall, McGraw-Hill, Revised 2nd Edition, 2005

References:

1. Advanced Microprocessors and Peripherals, A.K. Ray and K.M. Bhurchandi, McGraw-Hill, 3rd Edition, 2012

2. Microprocessors and Microcontrollers, N. Senthil Kumar, M. Saravanan, S. Jeevanathan, Oxford University Press, 1st Edition, 2010

3. Assembly Language Programming and Organization of the IBM PC, Ytha Yu and Charles Marut, McGraw-Hill, 2nd Edition, 1992

4. Modern Digital Electronics, R. P. Jain, McGraw-Hill, Fourth Edition, 2010

Online References:

Sr. No.	**Website Name**
1.	NPTEL – Microprocessors and Microcontrollers, Fundamentals of Microprocessors
2.	Intel Developer Zone
3.	CrisisNet for dataset insights (for project simulation)
4.	YouTube Channels: Neso Academy, Gate Smashers (Assembly & Interfacing)

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12212404	Integrated Lab 2	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-1	IAT-2	IAT-1+ IAT-2				
12212404	Integrated Lab 2	--	--	--	--	25	25	50

Lab Objectives:

1) Provide hands-on experience with 8086 assembly language programming.
2) Enable design and implementation of field-ready embedded systems for CrisisNet use cases.

3) Encourage innovation through project-based learning for social impact (SDG 11 & 13)
4) To develop foundational skills in Linux system administration, including user management, file operations, and process handling.
5) To introduce core DevOps principles and practices including DevSecOps, SRE, and NoOps
6) To equip students with practical skills in version control using Git and GitHub/GitLab, shell scripting for automation, configuration management, containerization.

Lab Outcomes:

Students will be able to-

1) Write and test modular assembly programs for real-time applications.
2) Interface and control peripherals like switches, sensors, and displays.
3) Simulate and implement real-time alerting and communication routines.
4) Use microprocessors to prototype miniaturized disaster-response tools.
5) Apply embedded techniques to CrisisNet-relevant project solutions.
6) Document, debug, and present a functional embedded prototype.

List of Experiments.

Sr No	List of Experiments	Hrs
01	Write, assemble, and verify basic 8086 assembly language programs for arithmetic and logical operations using an emulator/simulator.	2
02	Generate accurate time delays using loop instructions in 8086 and verify the timing behavior on simulation tools.	2
03	Simulate and verify the functioning of 8279 keyboard/display interface for visual output in alert systems.	2

04	Interface 8255 with LEDs and switches to simulate and verify parallel I/O operations used in disaster signal panels.	2
05	Simulate and verify ADC interfacing to read analog sensor values such as temperature or flood levels using a digital circuit simulator.	2
06	Simulate and test interrupt-driven input handling using 8259 to prioritize sensor events in a disaster response context.	2
07	Configure and simulate DMA-based data transfer using 8237 and verify efficiency in handling bulk alert data.	2
08	Design and implement a mini-project simulating a CrisisNet component such as a flood monitoring system or emergency alert terminal using microprocessor hardware or simulation tools.	2
09	Interface stepper motor or buzzer control logic (simulated) to 8255 to model movement or acoustic alerts in hazard-prone zones.	2
10	To develop foundational skills in Linux system administration, including user management, file operations, and process handling.	2
11	To introduce core DevOps principles and practices including DevSecOps, SRE, and NoOps.	2
12	To enable effective use of Git and GitHub/GitLab for version control and collaborative software development	2
13	To equip students with the ability to write and use shell scripts for automating system and administrative tasks	2
14	To train students in configuration management, containerization, and building CI/CD pipelines using tools like Ansible, Docker, and Jenkins.	2

15	To provide knowledge of secure DevOps practices, monitoring systems, and logging tools aligned with modern observability and SRE frameworks	2

Sr No	List of Assignments / Tutorials	Hrs
1	Assignment- 1: Assembly Programming for Disaster Alert Logic- Write and simulate an 8086 assembly program to detect a critical threshold and trigger an alert. **Tools:** Emulator like Emu8086 or GNUSim8086 **Deliverable:** Program listing, screenshot of output, explanation of logic used.	2
2	Assignment- 2: Design I/O Interface for Emergency Button Panel Using 8255- Simulate and design an emergency input/output control panel using the 8255 PPI. **Tools:** Proteus or any digital simulator **Deliverable:** Circuit diagram, 8255 control word, explanation of each port's function, screenshots of simulation.	2

Text Books:

1. The 8086/8088 Family: Design, Programming and Interfacing, John Uffenbeck, PHI Learning, 2nd Edition, 2002

2. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications, Walter A. Triebel and Avtar Singh, PHI Learning, 4th Edition, 2002

3. Microprocessor and Interfacing: Programming and Hardware, Douglas V. Hall, McGraw-Hill, Revised 2nd Edition, 2005

4. The Linux DevOps Handbook by Damian Wojsław & Grzegorz Adamowic.

5. How Linux Works: What Every Superuser Should Know (3rd Edition) by Brian Ward

References:

1. Advanced Microprocessors and Peripherals, A.K. Ray and K.M. Bhurchandi, McGraw-Hill, 3rd Edition, 2012

2. Microprocessors and Microcontrollers, N. Senthil Kumar, M. Saravanan, S. Jeevanathan, Oxford University Press, 1st Edition, 2010

3. Assembly Language Programming and Organization of the IBM PC, Ytha Yu and Charles Marut, McGraw-Hill, 2nd Edition, 1992

4. Modern Digital Electronics, R. P. Jain, McGraw-Hill, Fourth Edition, 2010.

5. The Linux Documentation Project. (n.d.). Linux System Administrator's Guide. Retrieved from https://www.tldp.org/LDP/sag/html/index.html

6. Medium. (2021). DevOps: Complete CI/CD Pipeline Setup. Retrieved from https://medium.com/@abhishekjain.wisdom/devops-complete-ci-cd-pipeline-setup-d64b410f3240

Online References:

Sr. No.	**Website Name**
1.	NPTEL – Microprocessors and Microcontrollers, Fundamentals of Microprocessors
2.	Intel Developer Zone
3.	CrisisNet for dataset insights (for project simulation)
4.	YouTube Channels: Neso Academy, Gate Smashers (Assembly & Interfacing)
5.	Linux From Scratch (linuxfromscratch.org)
6.	Awesome DevOps Books on GitHub

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30)
12212405	IDEA LAB - 4(Innovation Design Engineering and Apply)	1	--	2*	15	--	30	15	60	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-1	IAT-2	IAT-1+ IAT-2				
12212405	IDEA LAB - 4(Innovation Design Engineering and Apply)	--	--	--	--	50	50	100

Rationale :

Aligned with the National Education Policy (NEP) 2020, the institution emphasizes experiential, interdisciplinary, and project-based learning through the IDEA Lab—a central hub for hands-on innovation.

To strengthen the undergraduate research ecosystem, the institution has adopted a theme-based academic model aligned with UN SGD. Each semester features six curated problem statements based on local need and aligned with core subjects in the same semester, enabling students to apply classroom knowledge to real-world challenges. Every student selects one problem and develops an individual, subject-integrated solution—enhancing both academic understanding and research skills.

The IDEA Lab supports this initiative with facilities for design thinking, prototyping, and product development. Students maintain a project logbook throughout the semester to track their progress and reflections.

To ensure academic accountability, a two-tier assessment framework is implemented:

- Project Assessment based on standardized IDEA Lab rubrics.

- Subject-Based Term Work Assessment focused on the application of same-semester subject knowledge in the project.

Lab Objectives:

1. To promote experiential and project-based learning that bridges theoretical knowledge with real-world problem-solving.
2. To encourage interdisciplinary integration by enabling students to apply concepts from multiple subjects within a single cohesive project.
3. To develop innovation and design thinking skills through hands-on activities and iterative solution development.
4. To foster critical thinking and creativity by engaging students in open-ended problems

with multiple solution pathways.

5. To enhance communication, collaboration, and documentation skills essential for professional engineering practice.
6. To build an entrepreneurial and research mindset by guiding students to develop scalable, socially-relevant, and technically viable prototype

Lab Outcomes: Student will be able to

1. Recall and articulate key concepts from core and allied subjects relevant to the assigned project.
2. Explain the interdisciplinary nature of the problem and the role of each subject in addressing it.
3. Apply appropriate tools, techniques, and theoretical knowledge to develop project components.
4. Analyze problem constraints and user requirements to structure a feasible and efficient solution.
5. Evaluate multiple design options and justify the chosen solution based on technical and practical considerations.
6. Create a functional prototype or solution that demonstrates innovation, utility, and integration of interdisciplinary knowledge

1) Guidelines for IDEA Project

a) Project Guidelines (Interdisciplinary Project Execution in IDEA Lab)

- Each student works on an individual interdisciplinary project aligned with the semester theme.
- Faculty in-charges for the IDEA Lab are assigned according to the complexity of the project and the capacity of the respective departments.
- Faculty in-charges mentor both the academic and technical aspects, and track weekly

progress.

- Project assessment will be rubric-based, ensuring depth, innovation, documentation, and ownership.
- Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
- Faculty in-charges must attend relevant FDPs to ensure uniformity in mentoring and evaluation.

b) Guidelines for same semester Subject Concepts Applied within the Project

- Termwork for each subject will partially reflect how well a student applies subject-specific concepts in their interdisciplinary project.
- Internal assessment panel will collaborate to align project components with subject learning outcomes.

---

c) Role of Faculty In-Charges in IDEA Lab Projects

Faculty in-charges play a pivotal role in the success of interdisciplinary, theme-based projects under the IDEA Lab. Their responsibilities extend beyond technical supervision to include academic alignment, innovation facilitation, and active student engagement. Their key roles include:

1. **Motivating and Inspiring Students**
   - Encourage students to take ownership of their learning and projects.
   - Cultivate a mindset of curiosity, exploration, and social relevance.
   - Foster an environment where students feel empowered to take creative risks.
2. **Conducting Brainstorming and Ideation Sessions**
   - Organize structured brainstorming sessions at the start of the semester to help students define their problem statements and solution pathways.
   - Promote collaborative thinking, design exploration, and interdisciplinary integration.

3. **Arranging Guest Lectures and Expert Talks**
   - Identify and invite industry experts, researchers, and innovators for guest lectures aligned with the semester's theme or subject areas.
   - Facilitate exposure to real-world challenges, current trends, and future opportunities.
4. **Ensuring Uniqueness and Originality of Projects**
   - Actively review proposed ideas to ensure **no duplication of solutions** across students.
   - Encourage students to explore novel approaches, technologies, and perspectives.
5. **Promoting Discussion and Collaborative Learning**
   - Create platforms for students to present, discuss, and receive peer and mentor feedback.
   - Facilitate idea refinement through regular discussions and group engagement.
6. **Aligning Subject Content Beyond Syllabus**
   - Faculty in-charges must **align subject content beyond the syllabus of the same semester** with the **IDEA Lab theme and assigned problem statements**.
   - This ensures relevance, depth, and meaningful interdisciplinary integration.
7. **Same Semester Faculty Requirement**
   - Faculty in-charges must be teaching subjects in the **same semester** as the students' project to ensure seamless academic integration and contextual understanding.
8. **Monitoring and Documentation**
   - Oversee project logbook maintenance, milestone tracking, and submission of progress reports.
   - Provide ongoing feedback and ensure project alignment with learning outcomes.
9. **Coordination with Subject Faculty**
   - Work in collaboration with other subject faculty to help students embed theoretical and practical aspects of their coursework into the project.
   - Facilitate subject-term mapping and contribute to termwork assessment based on evidence.

2) Implementation Strategy

a) Project Implementation in IDEA Lab

Aspect	Implementation Strategy
Faculty in-charges	**Faculty in-charges assigned based on project nature and department capacity.**

Aspect	Implementation Strategy
Mentoring Role	**Faculty in-charges oversee academic/technical development, interdisciplinary integration, and timely documentation.**
Capacity Building	**Faculty in-charges undergo workshops on design thinking, innovation, assessment rubrics, and outcome-based mentoring.**
Assessment Contribution	**Faculty in-charges contribute to 25 marks allocated for the IDEA Lab project termwork. The remaining assessments are conducted by the external examiner.**
Recognition & Incentives	**Faculty in-charges receive workload credits or are formally acknowledged in performance reviews.**

---

b) Implementation of Subject-Term Work Mapping within Projects

Aspect	Implementation Strategy
Mapping Subject Outcomes	**Faculty in-charges align their content beyond syllabus with the student's project by coordinating with the assigned project guide.**
Independent Evaluation	**Internal assessment panel evaluate students based on their application of subject-specific concepts within the project. This contributes to a** separate 25 marks allocated for termwork based on subject **application.**
Evidence Sources	**Evaluation is supported by project logbooks, subject-specific deliverables (e.g., tools, simulations, models), and review presentation inputs.**
Outcome Assurance	**Ensures practical demonstration of subject understanding and its integration into the interdisciplinary solution.**

Implementation Notes:

- Guide faculty assess their course's contribution using specific evidence such as:

- Logbooks
- Subject-specific outputs (e.g., simulations, designs)
- Paper publications or review presentations

2) Guidelines for Assessment

Two-tier rubrics are applied independently to evaluate subject concept application and innovation within the project.

a) Assessment of IDEA Lab Projects (Individual Interdisciplinary Projects) (25 Marks)

Presentation-Based Assessment Structure (Total: 25 Marks)

Assessment Month	Weightage	Marks
Month 1 (Formative 1)	20%	5 marks
Month 2 (Formative 2)	40%	10 marks
Month 3 (Formative 3)	40%	10 marks

Rubric-Based Evaluation Criteria

Criteria	Month 1 (5)	Month 2 (10)	Month 3 (10)
Problem Understanding	Connects problem to subjects	Defines interdisciplinary scope	Demonstrates deep conceptual grasp
Subject Knowledge Application	Identifies relevant concepts	Applies principles in design	Integrates multiple subject areas correctly
Innovation & Design Thinking	Proposes creative idea	Develops and tests feasible solution	Final solution shows originality and utility
Documentation & Presentation	Logbook initiated, plan presented	Mid-design log and visuals	Final report and demo completed

Criteria	Month 1 (5)	Month 2 (10)	Month 3 (10)
Progress & Ownership	Meets deadlines, shows planning	Demonstrates self-motivation	Completes project independently with reflection

---

b) Term Work Assessment of Subject Concepts Applied in Projects (25 Marks)

Applicable to All Subjects Integrated with Interdisciplinary Projects

To reflect meaningful application of subject knowledge, each subject will be assessed through the following rubric:

Criteria	Marks	Description
Subject Knowledge Application	8	Depth and accuracy of concept integration into the project
Practical Design or Tool Usage	5	Use of subject-specific hardware/software/simulation/tools
Documentation	4	Quality and clarity of subject-related logs and reports
Viva/Presentation	4	Ability to explain subject's relevance and role in the project
Continuous Engagement	4	Evidence of consistent participation via logbooks and feedback

c) Total Assessment Structure

Component	Marks	Assessed By
Termwork – Project Execution	25 Marks	Project Guide
Termwork – Application of Subject Concepts	25 Marks	IDEA Lab Panel
Viva Voce (Final Evaluation)	50 Marks	External Examiner

R-2025- S.E Computer Engineering

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12412406	Full Stack Development	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+IAT-2					
12412406	Full Stack Development	--	--	--	--	--	25	25	50

Lab Objectives:

1. To learn the fundamentals of Java and OOP principles.
2. To build backend applications using Spring Boot and REST APIs.

3. To develop frontend interfaces using HTML, CSS, and JavaScript.
4. To connect and perform CRUD operations with databases.
5. To deploy full stack apps and practice DevOps basics.
6. To promote self-learning using Git, Docker, and modern tools.

Lab Outcomes:

1) Understand and apply Java programming fundamentals including OOPs concepts. (LO1)
2) Build a responsive frontend using HTML, CSS, and JavaScript. (LO1)
3) Develop backend services and REST APIs using Spring Boot. (LO2)
4) Integrate frontend and backend and perform data operations using databases. (LO3, LO4)
5) Demonstrate skills in deploying full stack apps using Docker and cloud platforms.(LO4)
6) Utilize GitHub, Postman, and modern tools for full stack development workflow. (LO5,LO6)

list of Experiments:

Sr No	List of Experiments	Hrs.
01	Write a Java program to demonstrate class, object, and constructors	1
02	Implement method overloading and overriding using inheritance	2
03	Demonstrate file handling (read/write to file) and exception handling in Java	1
04	Use Java Collections (List, Set, Map) to perform basic data storage operations	2
05	Create and manage threads in Java using Thread class and Runnable interface	1
06	Design a responsive web page using HTML5 and CSS3 with Bootstrap	2
07	Build an interactive form and validate inputs using JavaScript and DOM manipulation	1
08	Set up a Spring Boot project and expose a simple REST API using @RestController	1
09	Test the REST API using Postman (GET, POST, PUT, DELETE) and analyze HTTP responses	2

10	Connect Spring Boot to MySQL/PostgreSQL using Spring Data JPA and perform CRUD operations	1
11	Implement entity relationships (OneToMany, ManyToOne) using JPA annotations	2
12	Integrate frontend HTML/JS form with Spring Boot backend using fetch/axios	2
13	Dockerize a Spring Boot application and run it as a container	1
14	Deploy the backend app to Heroku and frontend (static) to Netlify/Vercel	1
15	**Capstone Project**: Build and present a complete full stack app (e.g., To-Do List, Student Portal, Inventory System)	2

Sr No	List of Assignments / Tutorials	Hrs
01	Describe primitive data types in Java and explain the use of if, switch, for, while, and do-while with examples.	1
02	Define and explain the four pillars of OOP: Encapsulation, Inheritance, Polymorphism, and Abstraction with real-life analogies.	1
03	Write a comparative analysis of ArrayList, HashSet, and HashMap in terms of structure, use cases, and performance.	1
04	Discuss types of exceptions, try-catch-finally structure, and basic file reading/writing using FileReader and BufferedWriter.	1
05	Explain how each technology contributes to building modern, interactive, and responsive websites.	1
06	Define REST APIs and describe how Controllers, Services, and Dependency Injection work together in a Spring Boot application.	1

07	Explain how CRUD operations are performed using JpaRepository and annotate entity classes for one-to-one/many relationships.	1
08	Write an overview of CI/CD pipelines, environment variables, Docker containers, and version control using GitHub.	1

Text Books:

1. **"Java: The Complete Reference" by Herbert Schildt**
2. **"Head First Java" by Kathy Sierra and Bert Bates**
3. **"HTML and CSS: Design and Build Websites" by Jon Duckett**
4. **"Pro Spring Boot 3" by Felipe Gutierrez**
5. **"Docker Deep Dive" by Nigel Poulton**

References:

1. Herbert Schildt – *Java: The Complete Reference*
2. Craig Walls – *Spring in Action*
3. Kirupa Chinnathambi – *Learning React*
4. Josh Long – *Spring Boot: Up and Running*

Online References:

Sr. No.	Website Name
1.	https://docs.oracle.com/javase
2.	https://spring.io – Official Spring Boot Docs
3.	https://reactjs.org – React Documentation
4.	https://getbootstrap.com – Bootstrap Docs

5.	https://www.w3schools.com – HTML/JS/CSS

Assessment:

- **Term Work Marks:** 25 Marks (Total marks) = 10 Marks (Experiment) + 10 Marks (Capstone Project) + 5 Marks (Attendance)
- **Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.**

Course Code	Course Name	Teaching Scheme (Contact Hours)			Teaching Scheme (Contact Hours Per Semester)					Total
		L	T	P	L	T	P	SL	Notional Learning Hour	
9951149WW	**Open Elective**	**2**	**-**	**-**	**30**	**-**	**-**	**30**	**60**	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-1	IAT-2	IAT-1+IAT-2					
9951149WW	**Open Elective**	--	--	--	--	--	25	--	25

Students must select a course for Open Elective from a bucket provided by the Rahul Education Group.

Term Work Assessment: 25 Marks (Total marks) = 15 Marks (Case Study/Activity) + 5 Marks (Assignments) + 5 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
98451410	Design Thinking	2	-	-	30	-	-	30	60	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-1	IAT - 2	IAT-1+IAT-2					
98451410	Design Thinking	20	20	40	60	2.5	--	--	100

Rationale :

Design Thinking is a human-centered approach that fosters creativity, empathy, and problem-solving. This course equips students with tools to understand user needs, generate innovative ideas, and develop practical solutions for real-world challenges.

Course Objectives:

1. Introduce the core principles and stages of the Design Thinking process.
2. Develop an empathetic mindset to understand user needs.
3. Frame clear problem statements using user-centered tools.
4. Apply ideation techniques to generate creative solutions.

5. Understand various types of prototypes and their role in product development.
6. Learn the process of testing and refining ideas based on feedback.

Course Outcomes:

On successful completion of the course learner will be able to:

1. Compare traditional and design thinking-based approaches to problem solving.
2. Define user personas using empathy and mapping techniques.
3. Frame actionable problem statements using POV and problem definition tools.
4. Apply ideation techniques to develop diverse solution sets.
5. Differentiate between low and high-fidelity prototypes and understand their usage.
6. Explain how to test prototypes based on desirability, feasibility, and usability..

Prerequisite: Fundamentals of communication and leadership skills.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Fundamentals of communication and leadership skills		
I	**Introduction to Design Thinking**	**Introduction to Design Thinking:** Definition, Comparison of Design Thinking and traditional problem-solving approach, Need for Design Thinking approach, Key tenets of Design Thinking, 5 stages of Design Thinking (Empathize, Define, Ideate, Prototype, Test)	4	CO1
		Self-learning Topics:	4	

		Design thinking case studies from various domains. **Link:** https://www.design-thinking-association.org/explore-design-thinking-topics/external-links/design-thinking-case-study-index		
II	**Empathy**	Foundation of empathy, Purpose of empathy, Observation for empathy, User observation technique, Creation of empathy map.	5	CO2
		Self-learning Topics: Creation of empathy maps **Link:** https://www.interaction-design.org/literature/topics/empathy-mapping **Case Study:** Designing MRI for Children – GE Healthcare	5	
III	**Define**	**S**ignificance of defining a problem, Rules of prioritizing problem solving, Conditions for robust problem framing, Problem statement and POV	5	CO3
		Self-learning Topics: Creating a Persona – A step-by-step guide with tips and examples **Link:** https://uxpressia.com/blog/how-to-create- persona-guide-examples	5	
IV	I**Ideate**	What is ideation? Need for ideation, Ideation techniques, Guidelines for ideation: Multi- disciplinary approach, Imitating with grace,	6	CO4

		Breaking patterns, Challenging assumptions, Looking across value chain, Looking beyond recommendation, Techniques for ideation: Brainstorming, Mind mapping		
		Self-learning Topics: How To Run an Effective Ideation Workshop: A Step-By-Step Guide **Link:** https://uxplanet.org/how-to-run-an-effective- ideation-workshop-a-step-by-step-guide-d520e41b1b96	6	
V	**Prototype**	Low and high-fidelity prototypes, Paper prototype, Story board prototype, Scenario Prototype.	4	CO5
		Self-learning Topics: IDEO Prototyping Toolkit	4	
VI	**Test**	5 guidelines of conducting test, The end goals of test: Desirability, Feasibility and Viability, Usability testing	6	CO6
		Self-learning Topics: Real-world Testing Examples from Stanford d.school or IDEO **Case Study:** IDEO's Shopping Cart Redesign – A classic innovation sprint (All stages)	6	
Total			30	

Textbooks:

1. Design Your Thinking: The Mindsets, Toolsets, and Skill Sets for Creative Problem-solving, Pavan Soni, Penguin, Random House India Private Limited
2. Design Thinking: Methodology Book, Emrah Yayichi, 2016
3. Handbook of Design Thinking: Christian Mueller-Roterberg, 2018

Reference books:

1. Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School, Idris Mootee, Wiley, 2013
2. Change by Design, Tim Brown, Harper Business, 2009

Online References:

Sr. No.	Website Name
1.	Design Thinking and Innovation by Ravi Poovaiah https://onlinecourses.swayam2.ac.in/aic23_ge17/preview
2.	Introduction to Design Thinking by Dr. Rajeshwari Patil, Dr. Manisha Shukla, Dr Deepali Raheja, Dr. Mansi Kapoor https://onlinecourses.swayam2.ac.in/imb24_mg37/preview
3.	Usability Testing https://www.interaction-design.org/literature/topics/usability-testing

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

Question paper format

- Question Paper will comprise of a total of **six questions each carrying 15 marks Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **Four questions** need to be answered

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
98431411	**Business Model Development**	2	-	-	30		-	30	60	**2**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -1	IAT-2	IAT-1+IAT-2					
98431411	**Business Model Development**	--	--	--	--	--	25	--	25

Rationale :

Entrepreneurship is the backbone of innovation and economic growth. In the 21st-century knowledge economy, students must be equipped not just with employment-ready skills but with a problem-solving and innovation mindset to become job creators. This course blends core entrepreneurship principles with business model design, startup financing, and digital business strategies.

Course Objectives:

1. To introduce students to entrepreneurship and its socio-economic relevance.

2. To enable understanding of different business structures and MSME significance.
3. To familiarize learners with funding mechanisms for startups.
4. To introduce the basics of IPR and ethical innovation practices.
5. To equip students with tools for building business models and MVPs.
6. To explore digital marketing and disruption in online business ecosystems.

Course Outcomes:

On successful completion of the course learner will be able to:

1. Define and explain entrepreneurship and its key types.
2. Analyze the legal and operational structure of business ownership.
3. Identify and compare startup funding models and sources.
4. Apply the fundamentals of IPR to protect entrepreneurial innovations.
5. Create comprehensive business models using BMC tools.
6. Evaluate and apply digital marketing techniques for online business models.

Prerequisite: Basic Design Thinking principles.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basic Design Thinking principles		
I	**Introduction to Entrepreneurship**	Definition, the role of entrepreneurship in the economic development, The entrepreneurial process, Women entrepreneurs, Corporate entrepreneurship, Entrepreneurial mindset.	5	CO1
		Self-learning Topics: Case studies: Henry Ford https://www.thehenryford.org/docs/default- source/default-document-library/default-	5	

		document-library/henryfordandinnovation.pdf?sfvrsn=0 The Tatas: How a Family Built a Business and a Nation by Girish Kuber, April 2019, Harper Business		
II	**Entrepreneurship Development**	Types of business ownerships: Proprietorship, Public and Private Companies, Co-operative businesses, Micro, Small and Medium Enterprises (MSME): Definition and role of MSMEs in economic development.	4	CO2
		Self-learning Topics:MSME Act and Udyam Registration portal **Case Study:** Amul – Co-operative innovation in dairy sector	4	
III	**Start-up financing**	**Start-up financing:** Cost and revenue models, Sources of start-up fundings: Angel investors, Venture capitalists, Crowd funding, Government schemes funding(e.g., SIDBI, Startup India).	5	CO3
		Self-learning Topics: Successful business pitching. **Case Study:** Shark Tank pitch breakdown.	5	
IV	**Intellectual Property Rights (IPR)**	Types of IPR: Patents, trademarks and copyrights, Patent search and analysis, Strategies for IPR protection, Ethics in technology and innovation.	5	CO4
		Self-learning Topics: Case Study: WIPO Patent Search Exercise	5	

V	**Business Model Development**	Types of business models, Value proposition, Customer segments, Customer relationships, Channels, Key partners, Key activities, Key resources, Prototyping and MVP.	6	CO5
		Self-learning Topics: The Art of the Start 2.0: The Time-Tested, Battle- Hardened Guide for Anyone Starting Anything by Guy Kawasaki **Case Study:** The Rise of Zomato – A Disruptive Food Delivery Platform	6	
VI	**Digital Business Management**	Digital Business models (Subscription, Freemium *etc*), Digital marketing: Search Engine Optimization (SEO), Search Engine Marketing (SEM), Social media and influencer marketing, Disruption and innovation in digital business.	5	CO6
		Self-learning Topics: Case study: Airbnb https://www.prismetric.com/airbnb-business-m	5	
Total			30	

Sr No	List of Assignments	Hrs
01	Presentation on case study of a failed business model	2
02	Presentation on case study of a woman entrepreneur	2

Textbooks:

1. Entrepreneurship: David A. Kirby, McGraw Hill, 2002
2. Harvard Business Review: Entrepreneurs Handbook, HBR Press, 2018
3. Business Model Generation; Alexander Ostlewalder and Yves Pigneur, Strategyzer, 2010
4. E- Business & E– Commerce Management: Strategy, Implementation, Practice – Dave Chaffey, Pearson Education

Reference books:

1. Entrepreneurship: New venture creation by David Holt, Prentice Hall of India Pvt. Ltd.
2. E- Business & E– Commerce Management: Strategy, Implementation, Practice – Dave Chaffey, Pearson Education

Online References:

Sr. No.	Website Name
1.	Startup India Portal – https://www.startupindia.gov.in
2.	Entrepreneurship by Prof. C Bhaktavatsala Rao https://onlinecourses.nptel.ac.in/noc20_mg35/preview
3.	Innovation, Business Models and Entrepreneurship by Prof. Rajat Agrawal, Prof. Vinay Sharma https://onlinecourses.nptel.ac.in/noc21_mg63/preview
4.	Sarasvathy's principles for effectuation https://innovationenglish.sites.ku.dk/model/sarasvathy-effectuation/
5.	WIPO Patent Search https://www.wipo.int/portal/en/index.html
6.	Airbnb Business Model Case https://www.prismetric.com/airbnb-business-model/
7.	BMC Tool https://www.strategyzer.com/canvas/business-model-canvas

Assessment:

Term Work: Term Work shall consist of at least 4 to 5 **Case Study Analysis , Presentation** based on the above list. Also, Term work Journal must include at least 2 to 3 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Case Study report)+ 5 Marks (Assignments) + 5 Marks (Attendance).