

Marathwada Mitra Mandal's College of Engineering

Karvenagar, Pune- 52

An Autonomous Institute Affiliated to SPPU



Curriculum for
Second Year B. Tech Computer Engineering
A.Y. 2026-27

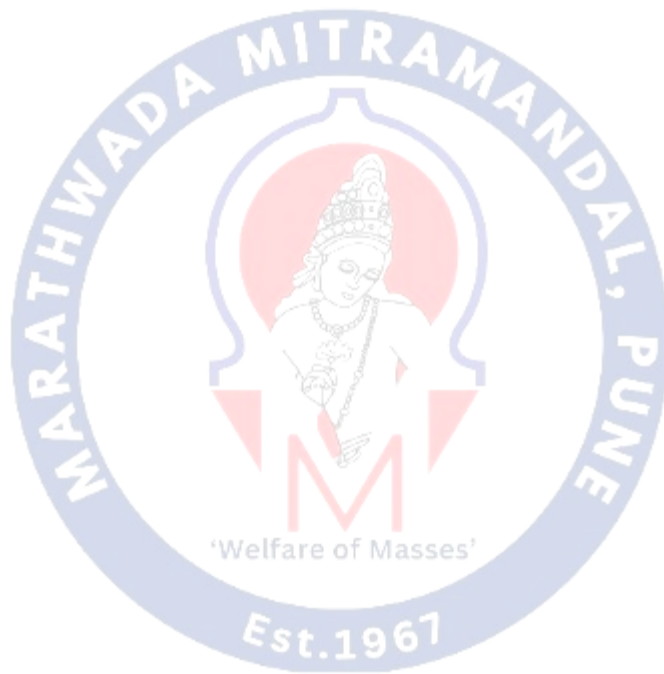
Permanently Affiliated to SPPU | Accredited with A++ grade by NAAC
Recipient of Best College award by SPPU | Accredited by NBA
Recognized under 2(f) and 12(B) of UGC Act 1956
www.mmcoe.edu.in

CONTENTS

Institute Vision and Mission	I
Department Vision and Mission	I
Knowledge and Attitude Profile (WK)	II
Program Outcomes (POs)	III
Program Educational Objectives (PEOs)	IV
Program Specific Outcomes (PSOs)	IV
Abbreviations	V
Curriculum Structure Sem-III & Sem-IV	VI
Semester-III Courses	1
Semester-IV Courses	32

Sr. No.	Description	Page
Semester- III		
CE24PCC201	Data Structures	2
CE24PCC202	Object Oriented Programming	4
CE24PCC203	Digital Electronics and Computer Organization	6
CE24PCC205	Data Structures Laboratory	8
CE24PCC206	Object Oriented Programming Laboratory	13
CE24CEF208	Project based Learning	17
CE24RMD210	Research Methodology & Publication	24
CE24EEM211	Business Ideation & Entrepreneurship	26
SH24VEC201	Environment Sustainability	30
Semester –IV		
CE24PCC251	Database Management Systems	33
CE24PCC260	System Programming and Operating System	35

CE24PCC253	Software Engineering	37
CE24PCC255	Database Management Systems Laboratory	39
CE24PCC261	System Programming and Operating System Laboratory	43
CE24VSE257	Web Technology	47
CE24VEC259	Universal Human Values	51
SH24AEC202	Professional Communication and Personality Development	54



Institute Vision

To be a globally renowned institution through excellence in engineering education for sustainable and holistic development

Institute Mission

M1: Empower students with cutting-edge technologies and global competencies

M2: Foster culture of research and entrepreneurial mindset

M3: Imbibe social and professional values

M4: Provide an inclusive environment for lifelong learning

Department Vision

To excel in computer engineering education for sustainable development.

Department Mission

M1: To develop globally competent professionals through interdisciplinary learning and Center of Excellence.

M2: To empower research, innovation, and entrepreneurial thinking

M3: To promote ethical values and holistic development

M4: To prepare graduates for lifelong learning and dynamic careers

Knowledge and Attitude Profile (WK)

WK1: A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.

WK2: Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.

WK3: A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.

WK4: Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.

WK5: Knowledge, including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.

WK6: Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.

WK7: Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.

WK8: Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.

WK9: Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Program Outcomes (PO)

P01: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

P02: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

P03: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

P04: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

P05: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

P06: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

P07: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

P08: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

P09: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

P010: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

P011: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Program Educational Objectives (PEO)

- PEO1:** Graduates will demonstrate strong foundational knowledge to address global and multidisciplinary challenges
- PEO2:** Graduates will engage in innovation, research, and entrepreneurship across multidisciplinary domains
- PEO3:** Graduates will practice ethical conduct, professional and social responsibility
- PEO4:** Graduates will exhibit lifelong learning and continuous skill enhancement

Program Specific Outcomes (PSOs)

A Graduate of the Computer Engineering Program will be able to

- PSO1:** Apply problem solving skills to design effective solutions in High Performance Computing, Artificial Intelligence and Cyber Security
- PSO2:** Develop advanced skill-based solutions using standard Software Engineering practices

Abbreviations	
BSC:	Basic Science Course
UG:	Undergraduate Programme
ESC:	Engineering Science Course
PCC:	Program Core Courses
PEC:	Program Elective Courses
MDM:	Multidisciplinary Minor Courses
OEL:	Open Elective
VSE:	Vocational & Skill Enhancement Course
AEC:	Ability Enhancement Course
EEM:	Entrepreneurship/Economics/Management
IKS:	Indian Knowledge System
VEC:	Value Education Course
RMD:	Research Methodology
CEP/FPR:	Comm. Eng. Project (CEP)/Field Project (FP)
PRJ:	Project
INT/OJT:	Internship/On-Job Training
CCC:	Co-Curricular Courses
IT:	Internal Tool
ET:	External Tool
ETE:	End -Term Examination
CIE:	Continuous Internal Evaluation
TW:	Term work
OR:	Oral Examination
PR:	Practical Examination
L:	Lecture
P:	Practical
T:	Tutorial
OL:	Online Teaching
ODL:	Open Distance Learning

Curriculum Structure Sem-III & IV

Second Year B.Tech Computer Engineering Semester-III

Course Code	Course Name	Course Type	Teaching Scheme (Hrs/week)			Examination Scheme						Credits			
			L	P	T	CIE	ETE	TW	PR	OR	Total	L	P	T	Total
CE24PCC201	Data Structures	PCC	3	-	-	40	60	-	-	-	100	3	-	-	3
CE24PCC202	Object Oriented Programming	PCC	3	-	-	40	60	-	-	-	100	3	-	-	3
CE24PCC203	Digital Electronics and Computer Organization	PCC	3	-	-	40	60	-	-	-	100	3	-	-	3
CE24MDM204	*Multidisciplinary Minor Theory	MDM	2	-	-	40	60	-	-	-	100	2	-	-	2
CE24PCC205	Data Structures Laboratory	PCC	-	2	-	-	-	-	50	-	50	-	1	-	1
CE24PCC206	Object Oriented Programming Laboratory	PCC	-	2	-	-	-	-	50	-	50	-	1	-	1
CE24MDM207	*Multidisciplinary Minor Laboratory	MDM	-	2	-	-	-	25	-	-	25	-	1	-	1
CE24CEP208	Project based Learning	CEP	-	4	-	-	-	50	-	-	50	-	2	-	2
CE24RMD210	Research Methodology & Publication	RMD	-	-	2	-	-	50	-	-	50	-	-	2	2
CE24EEM211	Business Ideation & Entrepreneurship	EEM	-	-	2	-	-	50	-	-	50	-	-	2	2
SH24VEC201	Environment Sustainability	VEC	-	-	2	-	-	50	-	-	50	-	-	2	2
Total			11	10	6	160	240	225	100	-	725	11	5	6	22

* Refer separate MDM Booklet

Second Year B.Tech Computer Engineering Semester-IV

Course Code	Course Name	Course Type	Teaching Scheme (Hrs/week)			Examination Scheme						Credits			
			L	P	T	CIE	ETE	TW	PR	OR	Total	L	P	T	Total
CE24PCC251	Database Management Systems	PCC	3	-	-	40	60	-	-	-	100	3	-	-	3
CE24PCC260	System Programming and Operating System	PCC	3	-	-	40	60	-	-	-	100	3	-	-	3
CE24PCC253	Software Engineering	PCC	3	-	-	40	60	-	-	-	100	3	-	-	3
CE24OEL254	* Open Elective	OEL	3	-	-	40	60	-	-	-	100	3	-	-	3
CE24PCC255	Database Management Systems Laboratory	PCC	-	2	-	-	-	-	50	-	50	-	1	-	1
CE24PCC261	System Programming and Operating System Laboratory	PCC	-	2	-	-	-	-	50	-	50	-	1	-	1
CE24VSE257	Web Technology	VSE	-	4	-	-	-	50	-	-	50	-	2	-	2
CE24VEC259	Universal Human Values	VEC	-	-	2	-	-	50	-	-	50	-	-	2	2
SH24AEC202	Professional Communication and Personality Development	AEC	-	-	2	-	-	50	-	-	50	-	-	2	2
Total			12	8	4	160	240	150	100	-	650	12	4	4	20

L- Lecture

ETE- End Term Examination

L : 1 Hr= 1 credit

P- Practical

TW- Term work

P: 2 Hr= 1 Credit

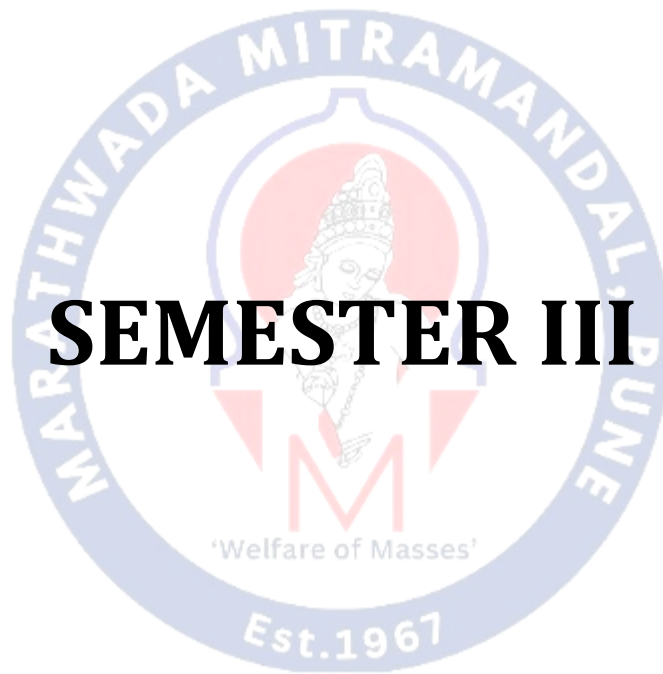
T- Tutorial

PR- Practical

T: 1 Hr= 1 Credit

CIE-Continuous Internal Evaluation

OR- Oral



Second Year B.Tech Computer Engineering														
Semester-III														
Course Code: CE24PCC201											Course Name: Data Structures			
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisite: Fundamentals of Programming Languages (CE24PCC101)														
Course Objectives: The course aims to: • Demonstrate the use of arrays for storing and accessing elements using index-based operations. • Explain the need for sorting techniques in optimizing data retrieval. • Construct and manipulate singly and doubly linked lists to perform insertion and deletion operations. • Design tree-based structures such as binary search trees for hierarchical data representation. • Apply tree and graph algorithms to find optimal paths in a network.														
Course Outcomes: After learning the course, the students will be able to: CO1: Implement basic operations on linear data structure CO2: Implement searching, sorting and hashing techniques for solving real life problems CO3: Develop programs using linked lists for solving problems CO4: Apply concepts of stack and queue data structures to real life problems CO5: Develop programs using nonlinear data structures for solving computing problems														
Unit	Contents													Duration (Hrs.)
1	Introduction to Data Structures: Data, Information, Knowledge, and Data structure, Abstract Data Types (ADT), Data Structure Classification, Characteristics of algorithm, Design tools of algorithm, analysis of algorithm using step count method and growth rate. Array: Sequential Organization, Array as an ADT, Operations, Storage Representation, Multidimensional Arrays, Sparse matrix representation, Simple and Fast Transpose of Sparse matrix, Time and Space tradeoff, Applications.													7
2	Searching, Sorting and Hashing: Searching: Linear and Binary Search Sorting methods: Bubble sort, Insertion Sort, Selection sort, Radix Sort. Hashing: Hash table, Hash functions, Properties of good hash function, Collision resolution strategies- open addressing and chaining, Applications.													7
3	Linked List: Static and Dynamic Allocation, Representation using dynamic memory management, Types: singly, Circular, Doubly, Primitive Operations - Create, Traverse, Search, Insert, Delete, Applications.													8
4	Stack and Queue: Stack: Concept, Representation using Sequential Organization, Operations (PUSH and POP), Applications: Polish notations, Expression Conversion and Evaluation, Recursion- concept, variants of recursion- direct, indirect, tail and tree.													8

	Queue: Concept, Representation using Sequential organization, Operations, Circular Queue, Double ended queue, Types of Double ended queue, Priority Queue, Applications.	
5	Tree and Graph: Tree: Tree Terminology, representation using array and linked list, Types, Tree traversal, Binary Search tree (BST): Insertion, deletion, Height Balanced Tree- AVL tree, B Tree, Applications. Graph: Graph Terminology, Storage representation, Adjacency matrix, adjacency list, Traversals-depth first and breadth first search, Minimum spanning tree: Prim's and Kruskal's algorithm, Applications.	10
Total Hours		40
Text Books		
1. Sartaj Sahni, Ellis Horowitz, "Fundamentals of Data Structures in C", Orient blackswan, 2nd Edition, 2010, ISBN 81- 7515-257-5. 2. Michael T. Goodrich, Roberto Tamassia, David M. Mount, "Data Structures and Algorithms in C++", ISBN-13 978-8126512607 Publisher WILEY INDIA EDITION, Publication date 1 January 2015.		
Reference Books		
1. E. Balagurusamy, "Object Oriented Programming with C++", Eighth Edition, ISBN-13 978-9389949186, McGraw Hill, 24 September 2020 2. Yashavant Kanetkar, "Let us C++", BPB Publications (16 September 2020), ISBN-10 : 9388176642, ISBN-13 : 978-9388176644. 3. G.A.V. PAI, "Data Structures and Algorithms, Concepts, Techniques and Applications", Tata McGraw-Hill, Volume1 1st Edition, 2017. ISBN-10: 0070667268/ ISBN-13: 978-0070667266. 4. Richard F. Gilberg & Behrouz A. Forouzan, "Data Structures, Pseudo code Approach with C", Cengage Learning India Edition, 2nd Edition, 2007, ISBN 10: 8131503143 ISBN 13: 9788131503140. 5. Y. Langsam, M. Augenstein, A. Tannenbaum, "Data Structures using C & C++", Pearson Education India, Second Edition, 2015, ISBN 10: 9332549311, ISBN 13: 978-9332549319.		
Online References		
1. NPTEL Course: Prof. Naveen Garg, IIT Delhi, "Data Structures And Algorithms" (Link: https://nptel.ac.in/courses/106102064) 2. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html 3. https://visualgo.net/en		

Second Year B.Tech Computer Engineering														
Semester- III														
Course Code: CE24PCC202							Course Name: Object Oriented Programming							
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
3		-	-	-	40	60	-	-	-	100	3		-	3
Prerequisite: Fundamentals of Programming Languages (CE24PCC101)														
Course Objectives: The course aims to: - Understand the concepts of data abstraction, encapsulation, and object oriented modeling. - Apply code reusability principles using inheritance. - Understand how virtual functions enable dynamic binding to achieve polymorphism. - Implement the concepts of exception and file handling. - Design generic classes with C++ templates and Standard Template Library.														
Course Outcomes: After learning the course, the students will be able to: CO1: Understand key concepts of object-oriented programming. CO2: Apply reusability principle using inheritance for object oriented design. CO3: Design solutions for real life problems using polymorphism concepts. CO4: Apply exception and file handling concepts in real world applications. CO5: Implement generic programming using template and STL in OOP applications.														
Unit	Contents										Duration (Hrs.)			
1	Introduction to OOP and Basic C++ Concepts Introduction to OOP: Overview of programming languages, Need and evolution of OOP, Key features of OOP, Applications of OOP. Classes and Objects: Declaring and defining classes, Specifying Member functions and data members, object creation, Memory allocation for objects, Access specifiers, Static members, friend functions. Function Prototype, Constructors and Destructors: Constructor types, Overloading of constructors, Destructor										8			
2	Inheritance and Pointers Inheritance: Need, Base and Derived classes, Types of inheritance, Access control in inheritance, Constructor and destructor in Derived Class, diamond problem (Virtual Base Class), Abstract class, Friend Class, Nested Class. Pointers: Declaration and initialization of pointers, indirection Operators, new & delete operator, Pointers to objects, this pointer, Arrays of Pointers, Function pointers, Pointers to Pointers, Pointers to Derived classes, Null pointer, void pointer.										8			
3	Polymorphism: Introduction, Types, Operator Overloading, Overloading Unary and Binary Operators, Function overloading, Data Conversion, Type casting, Keywords explicit and mutable. Run Time Polymorphism- Pointers to Base class, virtual function, pure virtual function and virtual table, virtual destructor, abstract base class.										8			
4	Files: Classes for file stream operation, Opening and closing a file, File modes, Error Handling functions in file, File Pointers and Their Manipulation, File										8			

	Operations on Characters, File Operations on Binary Files, Sequential input & output operations, Updating Random access file Exception Handling: Exception Handling mechanism using try, throw & catch, Divide by Zero exception, Multiple catch statements, re-throwing an exception, specifying exceptions, user defined exceptions, constructor, destructor and exception handling, exception and inheritance.	
5	Templates and Standard Template Library (STL) Templates: Introduction to Generic Programming, Function Template and Class templates, Template with multiple parameters, Overloading of template functions, Non-type template argument Standard Template Library (STL): Introduction to STL Components, Containers, Algorithms and Iterators, Application of Container classes: vector, list, map.	8
Total Hours		40
Text Books		
1. E. Balagurusamy, "Object -Oriented Programming with C++", McGraw Hill Education, Eighth Edition, Sept. 2020, ISBN-13 : 978-9389949186. 2. Robert Lafore, "OOP in C++", Pearson Publishing, 4th Edition, 2001, ISBN:0672323087 (ISBN 13: 9780672323089). 3. Yashwant Kanetkar, "Let Us C++", BPB Publications, ISBN: 9789388176644		
Reference Books		
1. Bjarne Stroustrup, The C++ Programming language, Third edition, 2008, Pearson Education. ISBN 9780201889543. 2. Deitel, C++ How to Program, 4th Edition, Pearson Education, 2002, ISBN:81-297-0276-2. 3. Herbert Schildt, C++ The complete reference, Eighth Edition, McGraw Hill Professional, 2011, ISBN:978-00- 72226805		
Online References		
1. NPTEL Course: Prof. Partha Pratim Das, IIT Kharagpur, "Programming in Modern C++" (Link: https://onlinecourses.nptel.ac.in/noc24_cs44/preview) 2. NPTEL Course: By Prof. Partha Pratim Das, IIT Kharagpur. "Programming in C++" (Link: https://onlinecourses.nptel.ac.in/noc21_cs02/preview)		

Second Year B.Tech Computer Engineering														
Semester- III														
Course Code: CE24PCC203					Course Name: Digital Electronics and Computer Organization									
Teaching Scheme (Hours/Week)					Examination Scheme							Credits		
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
3		-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisite: Basics of Electronics Technology (ET24PCC101), Basics of Electronics Technology Tutorial (ET24PCC102)														
Course Objectives: The course aims to: • Design simplified Boolean expressions using Karnaugh Maps to enhance logical circuit efficiency. • Construct combinational logic circuits such as multiplexers, decoders, and adders using Boolean logic. • Apply the concepts of sequential digital circuits to design systems using flip-flops, counters, and shift registers. • Implement assembly language programs by applying instruction sets and addressing modes. • Apply the concepts of memory hierarchy and I/O organization to understand and analyze computer system architecture.														
Course Outcomes: After learning the course, the students will be able to: CO1: Design K Map for minimization of Boolean Expressions. CO2: Apply the knowledge of combinational logic design. CO3: Apply the knowledge of sequential digital circuits. CO4: Apply addressing modes and instruction set to implement assembly language programs. CO5: Use the concept of I/O and memory organization for understanding computer systems.														
Unit	Contents										Duration (Hrs.)			
1	Combinational Circuits Minimization of Boolean function using K-map, Minimization of Product of Sum(POS) and Sum of Product(SOP) functions, don't care conditions,Introduction to Numbers and Codes, BCD, Gray, Excess-3 and their applications, Code conversion using logic gates, Integer and floating point number representation, Signed and unsigned numbers, arithmetic operations.										8			
2	Combinational Logic Design Realization of combinational functions like comparison, decoding, multiplexing, demultiplexing, Design of Half Adder and Full Adder, Design of Half Subtractor and Full Subtractor, BCD Adder, Look ahead and carry generator, Introduction to Carry Propagation Adder.										8			
3	Sequential Logic Design Flip-Flop: SR, JK,D,T, Preset and Clear, Master Slave JK Flip Flops, Truth Tables and Excitation tables, Conversion from one type to another type of Flop-Flop. Registers: SISO, SIPO, PISO, PIPO, Shift Registers, Bidirectional Shift Register, Ring Counter , Universal Shift Register Counters: Asynchronous Counter, Synchronous Counter, Modulus of the counter (IC 7490), Sequence Generator.										8			
4	Introduction to Computer Organization Function and structure of a computer, Functional components and Interconnection, Register organization, General purpose registers, Organization and optimization of microprogrammed control units. Instruction										8			

	cycle, type of instructions, types of operands, Instruction set, addressing modes. Instruction pipelining. Case study of 80386.	
5	Memory and Input/output Organization Memory devices and organization, Cache memory organization, principles, memory mapping, basics of virtual memory and address translation, performance characteristics, External memory devices and organization, Introduction to buses, bus organization, DMA organization. Case study of 80386.	8
Total Hours		40
Text Books		
1. R. P. Jain, 'Modern Digital Electronics', Tata McGraw-Hill, (5th Edition), (2022) 2. C. Hamacher, Z. Vranesic and S. Zaky, 'Computer Organization and Embedded Systems', McGrawHill, (5th Edition), (2017) W. Stallings, 'Computer Organization and Architecture - Designing for Performance', Prentice Hall of India, (10th Edition), (2016) 3. John P. Hayes, "Computer Architecture & Organization", 4. Douglas Hall, "Microprocessors & Interfacing", McGraw Hill, Revised 2 Edition, 2006 ISBN 0-07-100462-9 5. Intel 80386 Hardware Reference Manual 1986, Intel Corporation, Order no.: 231732-001, 1986. 6. A.Ray, K.Bhurchandi, "Advanced Microprocessors and peripherals: Arch, Programming & Interfacing", Tata McGraw Hill, 2004 ISBN 0-07-463841-6		
Reference Books		
1. John Yarbrough, "Digital Logic applications and Design", Cengage Learning, ISBN – 13: 978-81-315-0058-3 2. D. Leach, Malvino, Saha, "Digital Principles and Applications" , Tata McGraw Hill, ISBN –13:978-0-07-014170-4. 3. Anil Maini, "Digital Electronics: Principles and Integrated Circuits" , Wiley India Ltd, ISBN:978-81-265-1466-3. 4. Norman B and Bradley, "Digital Logic Design Principles", Wiley, ISBN:978-81-265-1258 5. Brey, Barry B, "8086/8088, 80286, 80386 and 80486 Assembly Language Programming", Prentice Hall, ISBN: 13: 9780023142475.		
Online References		
1. NPTEL series – nptel.ac.in/courses/117105080/ (Digital System Design by Prof. D. Roychoudhary, Dept. of Computer Science and Engineering, IIT Kh.) 2. Online Chapters – WilliamStallings.com/COA/COA8e.html		

Second Year B.Tech Computer Engineering														
Semester-III														
Course Code: CE24PCC205					Course Name: Data Structures Laboratory									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	2	-	-	-	-	-	-	-	50	50	-	1	-	1
Prerequisite: Fundamentals of Programming Languages Tutorial (CE24PCC102), Engineering Exploration Lab -1 (CE24VSE103)														
Course Objectives: The course aims to: • Apply linear data structures to perform insertion, deletion and traversal operations. • Implement sorting algorithms and hashing techniques to efficiently store and retrieve data. • Apply tree and graph algorithms to solve real-world problems for data analysis.														
Course Outcomes: After learning the course, the students will be able to: CO1: Implement linear data structures, searching, sorting, and hashing techniques to solve real-life problems. CO2: Design solution for real life problems using stack and queue data structures. CO3: Develop programs using nonlinear data structures for solving computing problems.														
List of Experiments														
Sr. No.		Name of the Experiment										Duration (Hrs.)		
1		Rainfall Tracking: Write a program to track rainfall data for 3 cities over 4 months. Using a 2D array, we can store the data, calculate the average rainfall for each city, and display the rainfall data in a tabular format. OR Temperature Tracker: Write a program for tracking daily temperatures of 3 cities for a week . The program calculates the average temperature for each day and for the week.										4		
2		Game Development: write a game development program that implements the Bubble Sort algorithm. The program will simulate a simple game where the player can input a set of numbers, and the numbers will be sorted using Bubble Sort to simulate a "level-up" scenario where the player's scores are sorted in ascending order. OR Organizing Cards in a Hand: Application: When playing card games, players often use an approach similar to insertion sort to organize their cards. They pick one card at a time and insert it into the correct position in their hand, maintaining a sorted sequence. Write a program that demonstrates how to organize (sort) cards in a hand using insertion sort										4		
3		Load Balancing: For example, imagine you have a set of servers that handle requests for a web application. The key to load balancing is using the hash value of a client's IP address or a request ID to determine which server should handle the request. The hash function is typically designed so that the data is evenly distributed										2		

	across the servers, ensuring that no single server is overloaded. Write a program of a load balancing system using hashing, where a basic hash table for mapping incoming requests to a set of servers.	
4	Simple Task Scheduler: Write a program that implements a simple task scheduler using a singly linked list. Each node in the linked list represents a task with its priority and execution time. Tasks are scheduled based on their priority, with higher priority tasks being executed first. Note: Use an array to store the URLs. Use a singly linked list to maintain the order of visits. When a new URL is visited: If the history is not full, add it to the head of the list and the array. If the history is full, remove the last node from the list (oldest visit) and replace it with the new URL. Update the head pointer and array index accordingly.	4
5	Parenthesis Checker: Write a program using a stack for push, pop, peek, and isEmpty operations. Write isBalanced() Function that iterates through the input expression, pushes opening brackets onto the stack. For closing brackets, it checks the top of the stack for a matching opening bracket. Ensures that all opening brackets are matched by the end of the traversal. Main Function: Accepts a string expression from the user. Uses isBalanced() to determine if the parentheses in the expression are balanced. OR Syntax Parsing in Programming Languages: Parsing expressions is a key step in many compilers and language processors. When a language's syntax requires parsing mathematical or logical expressions, converting between infix and postfix notation ensures that expressions are evaluated correctly. Accept an infix expression and show the expression in postfix form.	4
6	Coffee Shop Line (Simple Queue): Arrival: Customers arrive at the coffee shop and stand in line. Order Processing: The first customer in line gets their order taken, and the barista starts making the coffee. Serving: Once the first customer is served, they leave the queue, and the next customer in line moves forward to be served. Write a program to implement a simple queue OR Printer Spooler (Circular Queue): In a multi-user environment, printers often use a circular queue to manage print jobs. Each print job is added to the queue, and the printer processes them in the order they arrive. Once a print job is completed, it moves out of the queue, and the next job is processed, efficiently managing the flow of print tasks. Implement the Printer Spooler system using a circular queue without using built-in queues.	4
7	Web Crawling: a. Breadth First Search (BFS): Application: Indexing web pages for search engines. Example: A web crawler uses BFS to visit web pages systematically, starting from a seed URL and exploring links level by level. Nodes represent web pages. Edges represent hyperlinks. BFS ensures that pages at the same "depth" (distance from the starting page) are visited before moving to deeper levels. Write a program to simulate the indexing of web pages for a search engine using a Breadth-First Search (BFS) algorithm.	2

	b. Depth First Search (DFS): Application: Web crawlers use DFS to explore web pages systematically, following links and indexing content for search engines. Write a simple program to index web pages using Depth First Search (DFS). The program should simulate a web graph where pages are represented as nodes and hyperlinks as edges.	
Total Hours		24
Suggested Mini Projects List		
Note: The Mini Project should be implemented in a group of 3-4 students. Students can select any one from the following mini project titles. Students can select a mini project title of their own choice. Implementation can be in C/C++/Java		
P1	Postfix Expression Validator: A postfix expression is valid if it contains the correct number of operands and operators, and operators have sufficient operands to operate on. You can use a stack to validate a postfix expression by counting the operands and operators during traversal. Steps: 1. Traverse the postfix expression from left to right. 2. Push operands onto the stack. 3. When an operator is encountered, check if there are at least two operands to perform the operation. 4. If at the end, exactly one operand remains on the stack, the expression is valid.	6
P2	Social Network Analyzer: Analyze a social network (e.g., friendships in a class, followers on Twitter) using graph theory. Collect data on social connections. Represent the network as a graph. Nodes will be as individuals, edges as connections. Calculate centrality measures, i.e. degree, betweenness, closeness to identify influential individuals. Detect communities within the network. Visualize the network using graph visualization tools.	
P3	Huffman Coding: Implement Huffman coding for data compression. Create a Huffman tree based on the frequency of characters in a given text. Assign codes to characters based on their positions in the tree. Encode and decode messages using the Huffman codes	
P4	Library Book Catalog: Create a simple library book catalog system. Allow users to add new books to the catalog (with title, author, ISBN, etc.). Implement sorting options for the catalog (e.g., sort by title, author, ISBN, publication year). Allow users to search for books based on keywords. Implement efficient sorting algorithms to handle large catalogs.	
P5	Customer Service Call Center: Simulate a basic call center system. Use a queue to store incoming customer calls. Simulate call handling by assigning calls to available agents. Track call waiting times and agent idle times. Analyze call center performance and identify potential bottlenecks.	
P6	Hospital Emergency Room: Simulate a hospital emergency room patient triage system. Use a priority queue to prioritize patients based on their medical conditions (e.g., critical, urgent, minor).	

	Simulate patient arrival and treatment times. Analyze the impact of different priority queue implementations on patient wait times and overall system efficiency.	
P7	Undo/Redo Functionality: Implement basic undo/redo functionality for a text editor or drawing application. Use a singly linked list to store a history of user actions (e.g., typing, deleting, drawing). Allow the user to undo previous actions by traversing the list backwards. Allow the user to redo undone actions by traversing the list forwards.	
P8	Music Player Search: Develop a simple music player with a search feature. Create a library of music files with metadata (artist, album, song title). Implement search algorithms to find songs based on artist, album, song title, or keywords in song lyrics	
P9	Traffic Light Controller: Simulate a basic traffic light controller for a single intersection. Use a circular queue to store the sequence of traffic light phases (e.g., green for north-south, yellow for north-south, red for north-south, green for east-west, etc.). Cycle through the phases in the queue to control the traffic lights. Optionally, incorporate features like pedestrian crossing signals and adjust the timing of phases based on traffic volume.	
Total Hours		30
Guidelines for Laboratory Conduction: 1. All the assignments on all concepts are mandatory. 2. Assignments on all concepts covered on Linked list, stack, Queue, Tree and Graph are mandatory and should be implemented on coding platforms such as Leetcode, GitHub. 3. Operating System recommended: - 64-bit Open-source Linux or its derivative. 4. Programming tools recommended: - G++/GCC, Eclipse/Geany. Guidelines for Students: 1. The laboratory assignments are to be submitted by students in the form of a journal. 2. Journal consists of Vision Mission of Institute, Department, certificate, table of contents and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept, algorithm, time complexity, sample input and expected output, conclusion). Guidelines for Laboratory Assessment: 1. Continuous assessment of laboratory work is done based on overall performance and Laboratory performance of students. 2. Each Laboratory assignment assessment should assign grade/marks based on rubrics with appropriate weightage. 3. Suggested rubrics for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficiency, punctuality and neatness.		
Text Books		
1. Sartaj Sahni, Ellis Horowitz, "Fundamentals of Data Structures in C", Orient blackswan, 2nd Edition, 2010, ISBN 81- 7515-257-5. 2. Michael T. Goodrich, Roberto Tamassia, David M. Mount, "Data Structures and Algorithms in C++", ISBN-13 978-8126512607 Publisher WILEY INDIA EDITION, Publication date 1 January 2015.		
Reference Books		
1. E. Balagurusamy, "Object Oriented Programming with C++", Eighth Edition, ISBN-13 978-9389949186, McGraw Hill, 24 September 2020		

2. Yashavant Kanetkar, "Let us C++", BPB Publications (16 September 2020), ISBN-10 : 9388176642, ISBN-13 : 978-9388176644.
3. G.A.V. PAI, "Data Structures and Algorithms, Concepts, Techniques and Applications", Tata McGraw-Hill, Volume1 1st Edition, 2017. ISBN-10: 0070667268/ ISBN-13: 978-0070667266.
4. Richard F. Gilberg& Behrouz A. Forouzan, "Data Structures, Pseudo code Approach with C", Cengage Learning India Edition, 2nd Edition, 2007, ISBN 10: 8131503143 ISBN 13: 9788131503140.
5. Y. Langsam, M. Augensteinand, A. Tannenbaum,"Data Structures using C & C++",Pearson Education India, Second Edition,2015, ISBN 10: 9332549311, ISBN 13: 978-9332549319.

Online References

1. NPTEL Course: Prof. Naveen Garg, IIT Delhi, "Data Structures And Algorithms" ([Link: https://nptel.ac.in/courses/106102064](https://nptel.ac.in/courses/106102064))
2. <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
3. <https://visualgo.net/en>



Second Year B.Tech Computer Engineering														
Semester-III														
Course Code: CE24PCC206					Course Name: Object Oriented Programming Laboratory									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	2	-	-	-	-	-	-	-	50	50	-	1	-	1
Prerequisite: Fundamentals of Programming Languages (CE24PCC101)														
Course Objectives: The Course aims to: - Understand the principles of object-oriented programming (OOP) with core concepts of object-oriented design. - Acquire proficiency in the syntax and semantics of the C++ programming language. - Develop OOP programming skills to enhance advanced programming capability.														
Course Outcomes: After learning the course, the students will be able to: CO1: Apply key concepts of object-oriented programming to solve everyday challenges. CO2: Design solutions to real life problems using polymorphism concepts. CO3: Implement generic programming using template and STL in OOP applications.														
List of Experiments														
Sr. No.		Name of the Experiment										Duration (Hrs.)		
1		Operations on Bank Account Implement a class to illustrate working of a bank account of a customer that consists of following members Data members: i. Name of the customer ii. Account Type iii. Account Number iv. Available balance amount in the account Member Functions: i. Assign initial values using constructors. ii. Deposit an amount in account iii. Withdraw an amount iv. Display account detail										2		
2		Employee Information System Develop a program in C++ to create a database of an employee's information system containing the following fields: Name, employee ID, Department, Date of Joining, Contact address, Telephone number etc. Construct the database with suitable member functions to accept and print employee details. Make use of constructor types, destructor, static members, inline function and dynamic memory allocation using operators-new and delete.										4		
3		Educational Institution System Design an educational Institution that maintains a database of all staff members where the database is segregated into a number of classes having hierarchical relationships. Define all classes with suitable data members and required member functions to build the database and fetch individual details .										2		
4		Designing of Complex Number calculator Implement a class Complex which represents the Complex Number. Implement the following functions Using Operator Overloading: 1. Constructors (Include all constructor types) 2. Overload operator + to add two complex numbers using member function.										4		

	3. Overload operator * to multiply two complex numbers using friend function. 4. Overload operators << and >> to output and accept Complex Numbers	
5	Geometrical Shapes Area calculator Define the base class Shape with two data members of any numeric type that are employed to compute the area of respective shape. Add member functions to input the data members values and display the area in base class only. Derive two classes from base class namely, Triangle & Rectangle. Take the display function as virtual in base class and redefine it in the derived classes. Using above classes, write a program to accept parameters for triangle or rectangle and display the area Using Virtual Functions.	2
6	Exception Handling Design a program with the following functionalities: 1. A function to read two numbers(Double type) 2. A function to calculate division of these two numbers 3. A try block to detect and throw an exception when divide by zero condition occurs 4. Appropriate catch block to handle the exception thrown	2
7	Sequential Files Operations Develop a system that defines inventory class with suitable data members and member functions. Perform read and write operations using class objects.	4
8	Array operations Design a program with a template for sorting the accepted array and displaying it using integer or float type data. Implement any sorting type using Generic Programming.	2
9	Associative Container Map Implement a map associative container in which the keys will be the Permanent Registration Number (PRN) of the students and the values will be the name of the respective student. Develop a program that prompts the user for PRN and then it looks in the map, using the PRN as an index and returns the name of the student.	2
Total Hours		24
Suggested Mini Projects List Note: The Mini Project should be implemented in a group of 3-4 students. Students can select any one from the following mini project titles or can select mini project title of their own. Mini project implementation should be preferably in C++. Design and implement a mini-project for any application or use case using maximum features of Object Oriented Programming. Use of sequential files is mandatory.		
P1	Fitness Tracker Build an application to track workouts and daily activities where users can log their exercises, durations, and calories burned.Classes: Exercise, User, WorkoutPlan	6
P2	Quiz Application: Implement a quiz application where users can take quizzes, track scores, and see their performance.	
P3	Library Management System: Create a library management system that allows users to add, update, delete, and search for books. Implement functionalities like user authentication, book categorisation, and transaction tracking.	

P4	Medical Information System A Medical Information System is designed to store and organize patient medical information, such as demographic and clinical data, lab results, and other relevant information using a C++ programming language. The system is designed to allow easy expansion, and efficient and secure payments using object-oriented programming techniques to ensure the data is stored, retrieved, and managed efficiently. The system can use encryption and authentication techniques to ensure the data is kept secure and private for a particular patient	
P5	Weather application Design weather report system that holds a daily weather report with data members dayofmonth,high temp , low temp ,amount rain and amount snow. Display weather report using analysis of data.	
P6	Task Management Tool Develop a tool for managing lists of tasks, allowing users to add, edit, and delete tasks. Implement features like setting priorities, due dates, and categorization of tasks.	
P7	Payroll System Develop a system that offers a robust payroll application containing features such as salary calculations, deductions, and benefits for both employees and employers. The system facilitates the management of employee records, monitors hours worked, and generates comprehensive report.	
P8	Stock Management System Create a Stock Management System assisting businesses in effectively managing their inventory. It monitors current stock levels and provides functionalities for users to add, remove, and modify items within the inventory.	
Total Hours		30
Guidelines for Laboratory Conduction: 1. Assignments on all concepts are mandatory. 2. Assignments should be implemented on coding platforms such as GitHub, HackerRank, CodeChef. 3. Use of open-source software is encouraged. 4. Operating System recommended: - 64-bit Open-source Linux or its derivative. 5. Programming tools recommended: - G++/GCC, Eclipse. Guidelines for Students: 1. The laboratory assignments are to be submitted by students in the form of a journal. 2. Journal consists of prologue, certificate, table of contents and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, Theory- Concept, algorithm , program code with suitable input and output, conclusion). Guidelines for Laboratory Assessment: 1. Continuous assessment of laboratory work is done based on overall performance and Laboratory performance of students. 2. Each Laboratory assignment should be evaluated based on parameters with appropriate weightage. 3. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficiency, punctuality and neatness		
Text Books		
1. E. Balagurusamy, "Object -Oriented Programming with C++", McGraw Hill Education, Eighth Edition, Sept. 2020, ISBN-13 : 978-9389949186. 2. Robert Lafore, "OOP in C++", Pearson Publishing, 4th Edition, 2001, ISBN:0672323087 (ISBN 13: 9780672323089).		

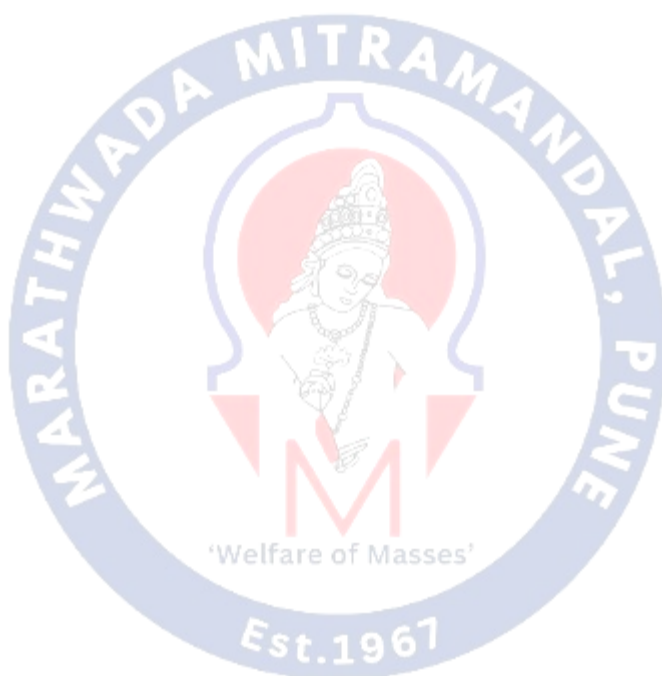
3. Yashwant Kanetkar, "Let Us C++", BPB Publications, ISBN: 9789388176644

Reference Books

1. Bjarne Stroustrup, The C++ Programming language, Third edition, 2008, Pearson Education. ISBN 9780201889543.
2. Deitel, C++ How to Program, 4th Edition, Pearson Education, 2002, ISBN: 81-297-0276-2.
3. Herbert Schildt, C++ The complete reference, Eighth Edition, McGraw Hill Professional, 2011, ISBN: 978-00-72226805

Online References

1. NPTEL Course: Prof. Partha Pratim Das, IIT Kharagpur, "Programming in Modern C++"
(Link: https://onlinecourses.nptel.ac.in/noc24_cs44/preview)
2. NPTEL Course: By Prof. Partha Pratim Das, IIT Kharagpur. "Programming in C++"
(Link: https://onlinecourses.nptel.ac.in/noc21_cs02/preview)



Second Year B.Tech Computer Engineering															
Semester- III															
Course Code: CE24CEP208											Course Name: Project based Learning				
Teaching Scheme (Hours/Week)					Examination Scheme						Credits				
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL	
-	4	-	-	-	-	-	50	-	-	50	-	2	-	2	
Course Objectives: The Course aims to: - Recognize and critically analyze environmental challenges encountered by local communities. - Explore and evaluate the role of technology in addressing societal needs. - Design and implement innovative solutions grounded in engineering principles to solve real-world problems. - Collaborate effectively in teams while contributing individually to the planning and development of projects.															
Course Outcomes: After learning the course, the students will be able to: CO1: Recall and explain relevant field knowledge, methods, and tools. CO2: Apply appropriate field research techniques to gather, analyze, and interpret data. CO3: Analyze field data critically to identify patterns, trends, and challenges, and evaluate their significance. CO4: Create innovative and sustainable solutions to address complex field-based problems, considering ethical and social implications.															
List of Lab Activities															
Sr. No	Activities													Duration (Hrs)	
1	Activity 1: Orientation of Field Project based Learning Activities Objective: Familiarize students with the field-based project, clarifying project goals, roles, tasks, and expectations. Tasks: - PBL Introduction - Brief overview of project-based learning. - Project Overview - Outline project scope, objectives, and activities. - Team Formation - Assign roles and responsibilities. - Timeline & Milestones - Present deadlines and key deliverables. - Fieldwork Prep - Discuss tools and methods for data collection. - Assessment Criteria - Explain evaluation standards. - Q&A - Address student queries. Expected Outcomes: - Clear understanding of PBL and project scope. - Defined team roles and responsibilities. - Preparedness for fieldwork. - Effective team collaboration. - Understanding of project timeline and deliverables.													2	
2	Activity 2:Field Visit & Problem Identification Objective: Evaluate the feasibility of the identified idea/concept from multiple aspects (technical, commercial, operational, and scheduling). Tasks: Students are expected to visit fields /company/organization to study specific Problem statements.													4	

	- Analyze the technical feasibility of the concept (requirements, challenges, technologies involved). - Assess the commercial feasibility (market demand, potential revenue, cost analysis). - Examine the operational feasibility (resources, processes, manpower). - Identify scheduling feasibility (timeframes, project milestones, deadlines). Expected Outcome: A comprehensive feasibility report that details the strengths, weaknesses, opportunities, and threats related to the concept.	
3	Activity 3: Information Gathering, Requirement Analysis, and Literature Survey Objective: Collect relevant information, analyze system requirements, and review existing literature to support the idea. Tasks : - Gather primary and secondary data on the concept and its potential solutions. - Conduct a requirement analysis for the system (functional, non-functional, and user requirements). - Perform a literature review of similar projects or systems, identifying key findings and gaps. Expected Outcome: A detailed collection of information, a requirements document, and insights from the literature that inform the system's design and development.	8
4	Activity 4: Review 1 Objective: Review the work done so far to assess the quality and completeness of the gathered information and analysis. Tasks : - Present the results of the feasibility evaluation and information gathering phase. - Provide an overview of the requirement analysis and the literature survey. - Receive feedback from peers or mentors on the direction of the project. Expected Outcome: Constructive feedback that can be used to refine the problem definition, research, and approach for the next stages of the project.	2
5	Activity 5: Preparing a System Architecture of the Proposed Idea Objective: Design a high-level architecture for the proposed system that outlines its components and interactions. Tasks: - Define the system's key modules and how they interact. - Create diagrams representing the architecture (e.g., block diagrams, data flow diagrams). - Consider scalability, security, and integration with existing systems or infrastructure. Expected Outcome: A system architecture document that clearly defines how the system will function and how its components interact with each other.	8
6	Activity 6: Designing the Proposed System and Specifying Hardware and Software Requirements Objective: Develop the detailed design of the system, specifying the hardware and software requirements. Tasks: - Design system components in detail, including database schemas, interfaces, and logic. - Identify the hardware specifications (e.g., servers, network devices) and software requirements (e.g., programming languages, frameworks). - Evaluate any dependencies or external systems needed for the implementation.	8
7	Activity 7: Implementation, Preparing Presentation, and Writing Report	7

	Expected Outcome: A design document that includes all hardware and software requirements, detailed designs for each component, and any integration needs.	
	Objective: Implement the system design, prepare the project presentation, and document the process in a report. Tasks: - Develop the system based on the final design, coding the necessary components using hardware and software specifications. - Create a presentation that highlights the system's features, functionality, and outcomes. - Write a comprehensive project report detailing the design, implementation process, and results. Expected Outcome: A working system prototype, a professional presentation for stakeholders, and a final report documenting the project from start to finish.	
8	Activity 8: Review 2 Objective: Perform a final review of the project to assess the overall success and lessons learned. Tasks : - Present the completed system, report, and presentation. - Evaluate the system's performance against the original objectives and requirements. - Discuss any challenges faced during the implementation and provide recommendations for future improvements. Expected Outcome: A final assessment of the system's effectiveness, a reflection on the project process, and recommendations for improvements or future work.	2
9	Activity 9: PBL Project Competition Objective: To showcase projects in the best possible way by setting up an engaging display, demonstrating its features, explaining our work clearly, answering questions confidently, and collecting feedback to improve solutions. Tasks : - Set up laptop, hardware, models, posters, charts, and other display materials. - Show key features, functionalities, and results to judges and visitors. - Respond confidently to queries from judges, faculty, and visitors. - Deliver a clear, time-bound presentation covering the problem, solution, methodology, results, and impact. Expected Outcome: A final assessment of the system's effectiveness, a reflection on the project process, and recommendations for improvements or future work.	4
Total Hours		45
Sample Field Project Based Learning (PBL) Topics		
P1	Software Development Smart Home Automation System Design a system that enables users to control and automate various home appliances remotely from any device, improving convenience, efficiency, and energy management. Task Management Application Develop an application to help small teams organize and manage tasks, projects, and deadlines efficiently, with features like to-do lists, calendar integration, and notifications.	8

	● E-commerce Website Build a platform that enables businesses to sell products online, providing an intuitive interface for product management, customer interaction, and secure transactions. ● Automated Resume Screening System Create an application that automatically processes resumes to filter and rank candidates based on their qualifications and suitability for specific job positions.	
P2	Embedded Systems ● Smart Traffic Management System Develop a system that optimizes traffic flow by detecting traffic conditions and adjusting signals dynamically to reduce congestion and improve traffic efficiency. ● Weather Monitoring System Create a system that collects environmental data (e.g., temperature, humidity, pressure) and provides real-time analysis to track weather conditions. ● Health Monitoring Wearable Design a wearable device that tracks vital health parameters and alerts users to potential health issues, offering remote monitoring features. ● Home Security System Build a system that enhances home security by integrating sensors, cameras, and remote monitoring, enabling users to oversee their property from anywhere.	
P3	Artificial Intelligence & Machine Learning ● Face Recognition System Develop a system that identifies individuals based on facial features, suitable for applications such as security or attendance tracking. ● Chatbot for Customer Service Create an intelligent conversational agent capable of handling customer inquiries, resolving issues, and managing support tickets in an efficient manner. ● Predictive Analytics for Stock Market Build a system that predicts market trends by analyzing historical financial data, aiming to provide actionable insights for investors. ● AI-based Personal Assistant Design an intelligent assistant that uses natural language processing to perform tasks like setting reminders, managing schedules, and providing information based on voice commands.	
P4	Data Science & Big Data ● Social Media Sentiment Analysis Develop a system that analyzes social media data to determine public sentiment about specific topics, products, or events, helping brands make informed decisions. ● Predictive Healthcare System Build a system that analyzes healthcare data to predict potential health issues and recommends preventative measures or treatments. ● Customer Behavior Analysis Design a tool to analyze consumer behavior, identify patterns, and suggest marketing strategies to improve customer engagement and sales. ● Data Visualization Dashboard Create a dashboard that visualizes large datasets, presenting key insights in an easily interpretable format for data-driven decision-making.	
P5	Cybersecurity	

	● Intrusion Detection System (IDS) Develop a system that detects and alerts about unauthorized access or potential cyberattacks in a network to enhance security. ● Password Strength Analyzer Build a tool that assesses the strength of passwords and provides recommendations to users for creating stronger and more secure credentials. ● Secure File Sharing System Design a system for securely sharing files, ensuring privacy through encryption, access control, and audit logs. ● Phishing Detection Tool Create a system that detects phishing attempts through emails or websites, helping users avoid security risks and fraud.	
P6	Networking ● Network Traffic Analyzer Develop a tool that monitors network traffic, identifies anomalies, and provides insights to optimize performance and detect potential issues. ● VPN System for Secure Browsing Build a system that ensures secure and anonymous internet access by encrypting user traffic and masking their IP address. ● Wi-Fi Coverage Mapper Design a tool that maps Wi-Fi signal strength across a given area and provides recommendations for improving signal coverage and connectivity. ● IoT-based Smart Farming System Develop a system that uses sensor data to monitor agricultural conditions (e.g., soil moisture, temperature) and improve farming efficiency through remote management.	
P7	Cloud Computing ● Cloud-based File Storage System Create a scalable, reliable cloud platform that allows users to store, synchronize, and share files securely across multiple devices. ● Cloud Resource Management Tool Develop a system for monitoring and optimizing the use of cloud resources, such as virtual machines, storage, and networks, to improve efficiency and reduce costs. ● Serverless Application Architecture Design an application architecture that eliminates the need for server management by utilizing serverless computing for scaling and cost efficiency. ● Disaster Recovery Plan for Cloud Infrastructure Create a solution to ensure business continuity by implementing cloud-based disaster recovery strategies, including backup and failover mechanisms.	
P8	Robotics ● Autonomous Delivery Robot Build a robot that can autonomously navigate and deliver items within a designated area, improving efficiency in logistics and transportation. ● Robot Arm for Precision Tasks Design a robotic arm capable of performing tasks that require high precision, such as assembly, sorting, or packaging, in industrial or research settings. ● Maze Solving Robot Develop a robot capable of autonomously solving mazes using algorithms, demonstrating problem-solving and decision-making capabilities. ● Robotic Hand for Prosthetics Create a functional robotic hand prototype that can be used as a prosthetic device, utilizing sensors to detect muscle signals for control.	

P9	Virtual Reality (VR) / Augmented Reality (AR) • Virtual Reality Learning Platform Develop an immersive learning platform that simulates real-world environments for educational purposes, offering interactive and engaging experiences. • Augmented Reality Navigation App Design an augmented reality application that provides real-time navigation assistance in indoor spaces like malls, airports, or museums. • VR-based Training Simulator Create a training simulator using virtual reality to provide hands-on, interactive training for fields such as healthcare, engineering, or aviation. • Interactive AR Shopping Experience Build an augmented reality app that allows users to visualize how products (such as furniture or clothing) would look in their own environment before purchase.	
P10	Human-Computer Interaction (HCI) • Gesture Recognition System Develop a system that enables users to control devices through hand or body gestures, offering a more intuitive method of interaction. • Voice-controlled System for Disabled Users Create a system that allows individuals with disabilities to interact with computers or devices using voice commands, improving accessibility. • Eye-tracking Software Design software that tracks eye movement to control devices or assist with reading, writing, or other activities, offering improved accessibility for users with disabilities. • Interactive Touchscreen for Education Build an educational tool that utilizes interactive touchscreens to engage students in learning activities, making education more interactive and dynamic.	
Total Hours		53
Guidelines For Faculty:-Assessment Method		
1. Project Milestones: Assessment based on the completion of project milestones (problem definition, design, coding, testing, deployment). 2. Lab Reports: Regular lab activity reports documenting the process, challenges, and solutions. 3. Final Project Presentation: A comprehensive project presentation at the end of the semester. 4. Peer Evaluation: Self-assessment and peer feedback for teamwork and collaboration. 5. Final Project Report: A detailed written report summarizing the entire project lifecycle, including design, implementation, and testing.		
Guidelines For Student		
Autonomous Learning: Students must learn to initiate tasks and generate ideas, not just imitate. They should think critically, define and analyze problems, and apply their learning to find solutions. They are responsible for their own learning. Active Participation: Students in PBL are expected to actively construct their knowledge through inquiry. They should not passively receive instruction but engage in the learning process by generating questions, analyzing information, and constructing solutions. Group Work and Interpersonal Skills: PBL requires students to work in groups, developing interpersonal skills such as effective listening, conflict resolution, and collaborative decision-making. Inquiry Skills: Students need to critically assess information by asking key questions such as:		

- How effective is the solution?
- How strong is the evidence?
- What justifies the chosen method?
- What is the supporting evidence?

Information Literacy: Essential for self-directed learning, this involves identifying when information is needed, locating relevant sources, and using that information to solve problems.

Collaborative Learning: PBL encourages students to work together to solve problems, share ideas, debate, and question each other's perspectives, enhancing learning through collaboration.

Interpersonal Skills for Group Success: Students should be aware of the need for skills like consensual decision-making, dialogue, and discussion to ensure effective problem-solving in group settings.

Note:

1. Students will be given the flexibility to choose their own problem statement.
2. A project basket containing problem statements is available to students.

Text Books

1. Problem Based Learning. by Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019
2. Project-Based Learning in Computer Science" Springer, 2020 by Brian H. Kessler "Software Engineering: A Practitioner's Approach" McGraw-Hill Education, 9th Edition (2014) by Roger S. Pressman
3. "Database System Concepts" McGraw-Hill Education 7th Edition, 2019 by Abraham Silberschatz
4. "Git Pocket Guide" O'Reilly Media ,2013 by Richard E. Silverman
5. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robert Capraro, Mary Margaret Capraro

Reference Books

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.
2. Project management core textbook, 2 Indian Edition , by Gopalan.
3. The Art of Agile Development. By James Shore & Shane Warden.

Online References

1. **PBLWorks:** <https://www.pblworks.org/>
2. **Edutopia:** <https://www.edutopia.org/>
3. **GitHub:** <https://github.com/>
4. **HackerRank:** <https://www.hackerrank.com/>
5. **LeetCode:** <https://leetcode.com/>
6. **Kaggle:** <https://www.kaggle.com/>

Second Year B.Tech Computer Engineering														
Semester- III														
Course Code: CE24RMD210					Course Name: Research Methodology & Publication									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	-	2	-	-	-	-	50	-	-	50	-	-	2	2
Prerequisite: -														
Course Objectives:														
• To introduce undergraduate students to the fundamentals of research and its importance in engineering innovation. • To develop awareness about research publications, patents, IPR, and research ethics. • To inculcate ethical research practices and scientific writing skills.														
Course Outcomes:														
After learning the course, the students will be able to:														
CO1: Apply the fundamental concepts of research in engineering to real-world engineering problems and demonstrate their significance.														
CO2: Apply basic research methodology and technical writing skills.														
CO3: Explain the process of research publications and intellectual property rights.														
CO4: Practice ethical standards in research activities.														
Contents														
1 Introduction and Importance of Research Research definition and meaning, Classification of research: basic, applied, experimental, analytical, and interdisciplinary research, Research in engineering and technology, Importance of research in innovation, sustainability, and societal development, Research problem identification and formulation														
2 Research Methodology and Technical Writing Introduction of Research methodology, steps involved in conducting research, Research design and methods of data collection, Basics of Data analysis and interpretation, Structure of a technical/research paper, Literature survey: purpose, sources, and review process, Use of digital research databases, Abstract writing, keywords, and conclusion writing, Referencing styles and citation tools														
3 Research Publications, Patents, and IPR Research publications: journals, conferences, review papers, and book chapters, Indexing and quality metrics, impact factor, h-index, citation count, Predatory journals and conferences. Intellectual Property Rights (IPR): Introduction, Types of IPR: patents, copyrights, trademarks, industrial designs, Patent process: prior art search, patent filing procedure, overview of patent drafting, Role of patents in engineering and entrepreneurship														
4 Research Ethics and Professional Conduct Research ethics and responsible conduct of research, Plagiarism: types, consequences, and prevention, plagiarism detection tools, acceptable similarity index, falsification, and duplication, authorship ethics and conflict of interest, ethical issues in sponsored and collaborative research, Overview of institutional ethics committees and ethical guidelines														

List of Tutorials		
Sr. No	Name of the tutorial	Duration (Hrs)
1	Identification and formulation of a research problem statement derived from relevant societal challenges, followed by a critical discussion on sustainable and feasible solutions.	4
2	Systematic literature review related to the identified research problem to understand the current state of knowledge, research gaps, and scope for innovation.	2
3	Development of a structured research methodology, including preparation of a detailed flowchart outlining the proposed research approach and experimental/analytical steps.	4
4	Hands-on activity on patent search and analysis, along with an introduction to the patent drafting process aligned with the proposed research methodology.	4
5	Activity on plagiarism detection tools and an in-depth discussion on research ethics, academic integrity, and responsible conduct of research.	4
6	Guided activity on writing a review paper based on the selected research topic, focusing on structure, critical analysis, and scholarly presentation.	4
7	Oral presentation and documentation of the research work, demonstrating the application of the proposed methodology, findings from literature, ethical compliance, and innovation potential.	4
Total Hrs		26
Text Books		
• Kothari C.R., "Research Methodology: Methods and Techniques", New age International Publications- 2024. • Dipankar Deb, Rajeeb Dey, Valentina E. Balas, "Engineering Research Methodology: A Practical Insight for Researchers", Springer Nature Singapore - 2019		
Reference Books		
• Ranjit Kumar, "Research Methodology: A Step by Step Guide for Beginners", SAGE Publications Pvt. Ltd - 2024		
Online References		
1. NPTEL Course: By Prof. Edamana Prasad, Prof. Prathap Haridoss, IIT Madras, "Research Methodology" (https://onlinecourses.nptel.ac.in/noc23_ge36/preview) 2. NPTEL Course: By Prof. Prathap Haridoss, IIT Madras, "Introduction to Research" (https://onlinecourses.nptel.ac.in/noc21_ge03/preview)		

Second Year B.Tech Computer Engineering														
Semester- III														
Course Code : CE24EEM211					Course Name: Business Ideation & Entrepreneurship									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	-	2	-	-	-	-	50	-	-	50	-	-	2	2
Pre-requisites: Communications, Problem Solving Abilities, and Analytical Thinking														
Course Objectives: Students should be able to: - Understand fundamental concepts of entrepreneurship, innovation, and the entrepreneurial mindset. - Apply creative and design thinking techniques to generate, refine, and prototype business ideas. - Develop and evaluate business models for technical, market, and financial feasibility using structured tools. - Recognize legal, organizational, and strategic aspects of starting and managing new ventures in the Indian context.														
Course Outcomes : After completing the course, the students will be able to: CO1: Identify business opportunities through problem and need analysis, and generate innovative solutions. CO2: Construct Business Model Canvas and Value Proposition Canvas for a proposed business idea. CO3: Evaluate the technical, market, and financial feasibility of a startup idea and prepare basic financial projections and go-to-market plans. CO4: Demonstrate types of entrepreneurship, causes of entrepreneurial failure, and relevant Indian business and company laws while proposing a viable venture.														
Unit	Contents										Duration (Hrs)			
1	Tutorial 1: Introduction to Entrepreneurship and Innovation Meaning and scope of entrepreneurship, Innovation and its role in economic development, Why become an entrepreneur? Motivation, rewards, and risks, Overview of course flow (from idea to venture launch).										2			
2	Tutorial 2: Business Ideation and Opportunity Identification Importance of business ideation, Sources of business ideas (personal experience, technology, trends, gaps in market, etc.), Opportunity identification: basic frameworks and checks, Short classroom exercise: list everyday problems and potential opportunities.										2			

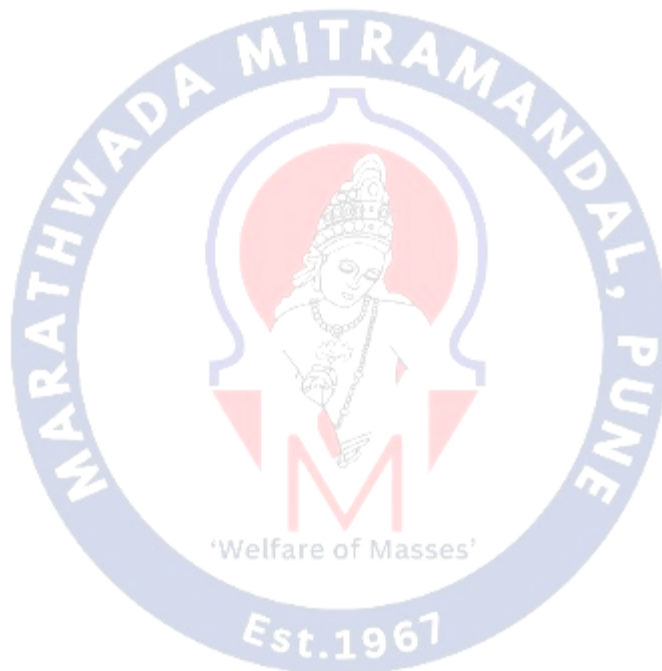
3	Tutorial 3: Problem Identification and Need Analysis Problem identification vs. solution jumping, Need analysis and customer pain points, Tools for understanding needs (observation, simple interviews, empathy maps), Mini-activity: write a problem statement for a chosen opportunity.	2
4	Tutorial 4: Entrepreneurial Mindset and Innovation Thinking Characteristics of an entrepreneurial mindset (risk-taking, resilience, creativity, proactiveness), Difference: Entrepreneur vs Intrapreneur; Entrepreneur vs Manager, Types of entrepreneurs: innovative, imitative, social, rural, family business entrepreneur (introduction only), Short reflection: assess your own entrepreneurial traits.	2
5	Tutorial 5: Creative Thinking Techniques I Brainstorming: rules, do's and don'ts, Mind mapping: visualizing ideas and connections Reverse thinking: looking at problems and assumptions in opposite ways Class activity: group brainstorming and mind map on a chosen theme.	2
6	Tutorial 6: Creative Thinking Techniques II and SCAMPER SCAMPER technique (Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Reverse). Applying SCAMPER to improve or reimagine an existing product/service Integrating techniques: brainstorm + mind map + SCAMPER Students refine one idea using SCAMPER.	2
7	Tutorial 7: Design Thinking Process Design Thinking overview: Empathize, Define, Ideate, Prototype, Test Deep dive into each stage with simple examples Linking design thinking to previous topics (problem identification, need analysis, creative techniques). Classroom mini-project: choose a problem and outline all five stages.	2
8	Tutorial 8: Convergent Thinking and Idea Evaluation Divergent vs convergent thinking, Screening and shortlisting ideas Feasibility analysis: concept and purpose. Types of feasibility: Technical feasibility, Economic feasibility, Market feasibility, Simple evaluation matrix for student ideas.	2
9	Tutorial 9: Business Model Development – BMC and Value Proposition Introduction to Business Model Canvas, Key partners, key activities, key resources, Value proposition, customer segments, channels, customer relationships, Revenue streams and cost structure. Introduction to Value Proposition Canvas: Customer jobs, pains, gains Products and services, pain relievers, gain creators. Activity: fill a simple BMC and VPC for a known company or student idea	2

10	Tutorial 10: Know Your Customers, MVP and Validation Customer segments and basics of market research (primary vs secondary) Revenue models (subscription, transaction, freemium, etc.) and cost structure basics Prototype and Minimum Viable Product (MVP) concept Idea validation techniques: Surveys and questionnaires, Feedback collection, Pilot testing / small experiments. Activity: design 5-7 survey questions for validating an idea	2
11	Tutorial 11: Entrepreneurship Types, Cases, and Failure Types of entrepreneurship in detail: Social entrepreneurship, Rural entrepreneurship, Family business entrepreneurship, The entrepreneurial decision process Concepts of entrepreneurial failure: Issues and reasons for failure (market, finance, team, regulation, execution, etc.), Essentials to avoid failure (planning, learning from feedback, pivoting, financial discipline). Case study of successful entrepreneurship in different sectors Case study of failure in entrepreneurship and key lessons	2
12	Tutorial 12: Financial Feasibility and Go-To-Market Financial planning for startups: why it matters Components of a basic financial plan: Sales forecast, Cost estimates Break-even concept (at introductory level) Building simple financial projections (revenue, expenses, profit) Go-To-Market (GTM) strategy basics: Target market, positioning, channels, Building and positioning the brand, Basics of digital marketing plan (social media, content, SEO at a high level).	2
13	Tutorial 13: Company Laws and Legal Aspects in India Types of business organizations in India: Sole Proprietorship, Joint Hindu Family Business, Partnership, Private Limited Company, One Person Company, Limited Liability Partnership (LLP).	2
Total Hours		26 Hrs
Text Books		
S. S. Khanka, <i>Entrepreneurship Development</i>, S. Chand Publishing, Latest Revised Edition, 2020. Poornima M. Charantimath, <i>Entrepreneurship Development and Small Business Enterprises</i>, Pearson Education India, 2nd Edition, 2018. Robert D. Hisrich, Michael P. Peters, Dean A. Shepherd, <i>Entrepreneurship</i>, McGraw-Hill Education, 11th Edition, 2023.		
Reference Books		
Kenneth R. Hall, <i>Entrepreneurship for Engineers</i>, Pearson Education, 1st Edition, 2012.		

2. **Thomas N. Duening, Robert D. Hisrich, and Michael A. Lechter**, *Technology Entrepreneurship: Taking Innovation to the Marketplace*, Academic Press (Elsevier), 3rd Edition, 2015.
3. **Alexander Osterwalder, Yves Pigneur, Gregory Bernarda, and Alan Smith**, *Value Proposition Design: How to Create Products and Services Customers Want*, John Wiley & Sons, 1st Edition, 2014.

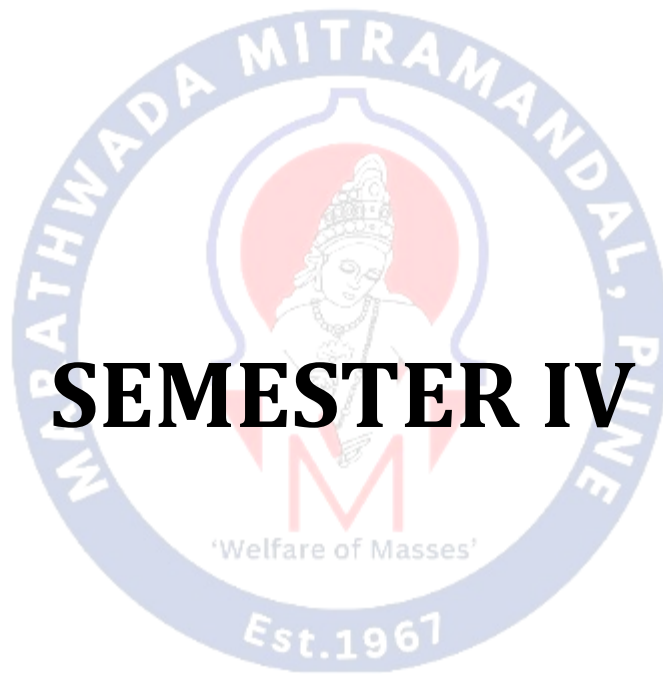
Online References

- <https://www.startupindia.gov.in/content/sih/en/reources/online-courses.html>
- <https://wadhwanifoundation.org/startup-funding-essentials/>



Second Year B.Tech Computer Engineering														
Semester- III														
Course Code: SH24VEC201										Course Name: Environment Sustainability				
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	-	2	-	-	-	-	50	-	-	50	-	-	2	2
Prerequisites: A basic understanding of environmental science, geography, or social studies														
Course Objectives: The course aims to: - Understand the foundational concepts of sustainability, including its social, environmental, and economic dimensions, and the global need for sustainable development. - Gain insights into the 17 United Nations Sustainable Development Goals (SDGs) and the role of technology, international cooperation, and policy in achieving them. - Identify and explain different types of pollution (air, water, solid waste), their sources and impacts, and sustainable approaches to waste management including the 3Rs (Reduce, Reuse, Recycle) and zero-waste concepts. - Evaluate global and local environmental challenges, including climate change, global warming, ozone depletion, and resource degradation														
Course Outcomes: After completion of the course, the students can: CO1: Describe the fundamental ideas of environmental sustainability. CO2: Examine sustainability's social, economic, and environmental facets. CO3: Analyse how human actions affect Environmental systems. CO4: Propose strategies for sustainable development in various sectors.														
Contents														
1	Introduction to Environmental Sustainability: Global environmental issues, Foundations of Environmental Sustainability. Sustainability: Challenges, potential; role of technology, policies.													
2	Environmental Pollution and Waste Management: Air Pollution: Sources and Effects of Air Pollution. Water Pollution: Sources of Water Pollution, Sustainable Wastewater Treatment. Solid Waste zero Waste Concepts (3R Concept).													
3	Climate Change and Energy: Understanding Climate Change, Science and impacts of climate change Climate modeling and adaptation, Sustainable Energy Solutions, Renewable energy technology, Efficient energy system and conservation													
4	Sustainable Development Practices: Sustainability in Cities, infrastructure and cities that are sustainable, green buildings and smart cities.													
List of Tutorial (Any 8)														
Sr. No.	Name of the Tutorial											Duration (Hrs.)		
1	Examine the river pollution in Pune.											4		
2	Prepare a climate change strategy for your town, city, or building.											4		
3	Initiate a campaign to bring emphasis to a sustainability concern.											4		

4	Examine The Current Environmental Issue	2
5	Examine a successful sustainability attempt from the real world	2
6	Home energy audit: Have students make a list of all the appliances and light bulbs in their house. How much energy does their house use if all the lights are on for 4 hours per day? If their appliances are on for 2 hours per day? How much energy could they save if they switched to energy-efficient appliances or lightbulbs?	2
7	Use recycled material in art projects: recycled materials can make beautiful art projects such as jewelry, planters, and birdhouses. Incorporating materials that would otherwise be thrown away into art projects can show your students how to find new uses for these items.	2
8	Life cycle: One way to show students what happens when you put something in the trash versus recycling or reusing the object is to do a life cycle analysis. This is a flowchart that shows the environmental impacts of an object, from extracting the raw materials to decomposition and everything in between. When something is put in the trash instead of being reused or recycled, the life cycle assessment will show a bigger environmental impact. When something is reused or recycled, the environmental impact is less because raw materials don't need to be extracted to create something new.	2
9	Go on a field trip: Visit your local landfill, recycling center, or a nearby composting facility where the students can see firsthand what is happening to waste and learn about the lifecycle of waste and its effect on the environment.	2
10	Compare carbon footprints of different materials used in construction or manufacturing in India.	2
Total Hours for 08 Activities		26
Text Books		
1. Sharma, "P.D. Ecology and Environment", Rastogi Publications, 2020. 2. Kaushik, C.P., Kaushik, Anubha, "Perspectives in Environmental Studies New Age", International Publishers, 2021. 3. Agarwal, S.K. "Environmental Management", APH Publishing Corporation, 2005.		
Reference Books		
1. Kalam, A.P.J. Abdul, Srijan Pal Singh, "Target 3 Billion Innovative Solutions Towards Sustainable Development", Penguin Books, 2011. 2. Kolbert, Elizabeth, "The Sixth Extinction: An Unnatural History", Henry Holt and Co., 2014.		
Online References		
● NPTEL Course : Prof. Trupti Mishra, IIT Bombay "Business and Sustainable Development" [https://nptel.ac.in/courses/110101153] ● NPTEL Course :By Prof. Brajesh Kumar Dubey , IIT Kharagpur "Sustainable Engineering Concepts And Life Cycle Analysis" [https://onlinecourses.nptel.ac.in/noc23_ce90/preview]		



Second Year B.Tech Computer Engineering														
Semester-IV														
Course Code: CE24PCC251						Course Name: Database Management Systems								
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-		100	3		-	3
Prerequisite: CE24MDM204:Mathematical Foundation for GenAI, CE24PCC201: Data Structures														
Course Objectives: The course aims to • Demonstrate database concepts, design principles, and ER/EER modeling. • Solve SQL queries and PL/SQL subroutines for efficient data processing. • Apply normalization techniques for designing well-structured relational databases • Explore database transactions, concurrency control methods, and recovery mechanisms. • Analyze NoSQL database models and their role in managing unstructured data.														
Course Outcomes: After learning the course, the students will be able to: CO1: Demonstrate the fundamentals of DBMS,ER/EER modeling. CO2: Develop SQL queries and PL/SQL subroutine to manage and manipulate relational data CO3: Apply normalization techniques for database design. CO4: Analyze transaction management concepts and concurrency control techniques for reliable database systems. CO5: Evaluate NOSQL database models and compare emerging database technology.														
Unit	Contents										Duration (Hrs.)			
1	Introduction to Database Management Systems and ER Model Introduction, Purpose of Database Systems, Database-System Applications, View of Data,Database Languages, Database System Architecture and Structure, Data Models. Database Design and ER Model: Entity, Attributes, Relationships, Constraints, Keys, Design Process, ER Diagram, Design Issues, Extended E-R Features, converting ER and EER diagrams into tables. Case Study: Study of Architecture of any DBMS like Oracle or MySQL. Design a database schema										9			
2	SQL and PL/SQL SQL: Characteristics and Advantages,SQL Data Types and Literals, SQL Commands: DDL, DML, DCL, TCL, Select queries, String operations, JOINS,Setoperations,SetMembership,Views,Indexes,Sequences,Aggregate,Nested Subqueries. PL/SQL: Stored Procedures, Functions, Cursors,Triggers. Case Study : Design and implement a Student Course Management System using SQL and PL/SQL to manage students, courses, and faculty members efficiently. The system should store and retrieve relevant data, ensuring integrity, security, and performance optimization.										9			
3	Relational Database Design & Normalization Relational Model: Basic concepts, Attributes and Domains, CODD's Rules. Relational Integrity: Domain, Referential Integrity, Enterprise Constraints, Database Design: Features of Good Relational Designs ,Functional Dependencies,1NF,2NF,3NF,BCNF.										9			

	Case study: Design and Optimization of a Relational Database for a University Management Systemright-linear grammar, Applications of grammar	
4	Database Transaction Management and Concurrency Control Basics of Transaction Management,ACID property, Schedules, Serializability, Locking and timestamp-based schedulers, Concurrency control,multi-version and optimistic Concurrency Control schemes, Database recovery. Case study: Design Online Shopping Cart Transaction Management In an e-commerce platform, multiple users simultaneously add, update, and purchase products. To ensure data consistency and reliability, the system must handle concurrent transactions effectively..	9
5	NoSQL and Emerging Database Technologies Distributed Databases: Features, advantages, disadvantages, CAP theorem, Types of Data: Structured, unstructured, Semi-structured, NoSQL Overview: Introduction, need, features; ACID vs BASE, Types of NoSQL Databases: Key-value, document, graph, wide-column stores. MongoDB Essentials: CRUD operations, indexing, aggregation, MapReduce. Emerging Database Technologies: XML,Mobile Database Case study: Study NoSQL Database Selection for a Social Media Platform	9
Total Hours		45
Text Books		
1. Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", McGraw Hill Publishers, ISBN 0-07-120413-X, 7th edition 2. Connally T., Begg C., "Database Systems", 4th Edition, Pearson Education, 2002, ISBN 8178088614 3. D TEditorial Services "BIG DATA Black Book", Dreamtech Press ISBN 13 : 978935119931		
Reference Books		
1. C J Date, "An Introduction to Database Systems", Addison-Wesley, ISBN: 0201144719 2. S.K.Singh, "Database Systems: Concepts, Design and Application", Pearson Education,ISBN 978-81-317-6092-5 3. Kristina Chodorow, Michael Dierolf, "MongoDB: The Definitive Guide", O'Reilly Publications, ISBN: 978-1-449-34468-9 4. Adam Fowler, "NoSQL For Dummies", John Wiley & Sons, ISBN-1118905628 5. Kevin Roebuck, "Storing and Managing Big Data - NoSQL, HADOOP and More",Emereo Pty Limited, ISBN: 1743045743, 9781743045749 6. Joy A. Kreibich, "Using SQLite", O'REILLY, ISBN: 13:978-93-5110-934-1 7. Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB Publications ISBN: 9788176569644, 9788176569644 8. Seema Acharya, "Demystifying NoSQL", Wiley Publications, ISBN: 9788126579969		
Online References		
- SQL and Relational Theory a. (How to Write Accurate SQL code), C.J. Date, O'REILLY Publication. - SQL A Beginner's Guide, Andy Oppel, Robert Sheldon, McGraw Hill Publication - NPTEL Course: Prof. Sreenivasa Kumar, IIT Madras, "Introduction to Database Systems" https://nptel.ac.in/courses/106106220 - NPTEL Course: Prof. Samiran Chattopadhyay,IIT Kharagpur, "Data Base Management System" https://nptel.ac.in/courses/106105175 - Mongodbcourse: https://www.mongodb.com/resources/basics/databases/nosql-explained		

Second Year B.Tech Computer Engineering														
Semester-IV														
Course Code: CE24PCC260					Course Name: System Programming and Operating System									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisite: CE24PCC203: Digital Electronics & Computer Organization														
Course Objectives: The course aims to: 1. Analyze the role of program translation tools. 2. Develop the ability to solve process scheduling and synchronization problems. 3. Apply memory and file management techniques for efficient resource utilization.														
Course Outcomes: After learning the course, the students will be able to: CO1: Identify design process of two-pass assembler. CO2: Analyze roles of system softwares macros, linkers and loaders in program translation. CO3: Compare different Process Scheduling algorithms. CO4: Solve Classical Synchronization Problems using semaphores and mutex. CO5: Evaluate memory management techniques.														
Unit	Contents										Duration (Hrs.)			
1	Systems Programming: Introduction, Types of software: system software and application software, Evolution of components of Systems. Assembly Language Programming: Assembly Language statements, A simple Assembly scheme, Pass Structure of Assembler. Design of two pass Assembler: Processing of declaration statements, Assembler Directives and Imperative statements, Advanced Assembler Directives, Intermediate code forms, Pass I and Pass II of Two Pass Assembler. Introduction to Compilers: Phases of Compiler with one example, Comparison of Compiler and Interpreter. Case Study: Role of System Software in Modern Operating Systems										9			
2	Macro Processor: Introduction, Features of a Macro facility: Macro instruction arguments, Conditional Macro expansion, Macro calls within Macros, Macro instructions, Defining Macro, Design of two pass Macro processor. Linker and Loader: Introduction, Loader schemes: Compile and Go, General Loader Scheme, Absolute Loaders, Subroutine Linkages, Relocating Loaders, Direct linking Loaders, Overlay structure, Self-relocating programs, Linker- Introduction, Types of linking. Case Study: Performance Impact of Macro Expansion in System Programs										9			
3	Introduction to Operating System Introduction to Operating System: Evolution, Services, Functions, Types, Virtualizing The CPU, Virtualizing Memory, Concurrency, Persistence. Process Management: Process, States: 5 and 7 state model, Process Control Block(PCB), Threads, Thread lifecycle, Multithreading Model, Process Scheduling Types: Preemptive, Non-preemptive, Schedulers, Scheduling Algorithms: FCFS, SJF, RR, Priority, Completely Fair Scheduler, Multilevel Feedback Queue. Case Study: Scheduling and Synchronization in xv6										9			
4	Synchronization and Concurrency Control Synchronization Vs. Concurrency Control, Critical Section Problem, Mechanisms, Classical Synchronization Problems Deadlocks: Introduction to Deadlock,										9			

	Conditions for Deadlock, Resource Allocation Graph (RAG), Wait-for Graph (WFG), Detection, Prevention, Avoidance - Banker's Algorithm, Recovery. Case Study: CPU Virtualization	
5	Memory Management and File Management Memory Management : Concepts, requirements, Memory Partitioning, Fragmentation, TLB, Paging, Segmentation, Placement Strategies, Virtual Memory: Concept, Swapping, Page Replacement Policies: FIFO, LRU, Optimal. Swapping issues: Thrashing File Management: Files, File Systems, File structure. File Organization and Access Methods, Allocation Methods, Directories, File Sharing, Record Blocking, Introduction to FAT16, FAT32 and NTFS Case Study: I/O Management and The xv6 filesystem	9
Total Hours		45
Text Books		
4. John J. Donovan, Systems Programming, McGraw-Hill, 1st Edition , ISBN-10: 0074604821 • ISBN-13: 978-0074604823 5. Dhananjay M. Dhamdhare, Systems Programming and Operating Systems (Second Revised Edition), Tata McGraw-Hill Education, 2nd Edition, 1999, ISBN-10: 0074635794 • ISBN-13: 978-0074635797 6. Operating Systems: Three Easy Pieces, Andrea Arpaci-Dusseau and Remzi Arpaci-Dusseau, Arpaci-Dusseau Books, 1st Edition, 2012, ISBN-10: 0985673529 • ISBN-13: 978-0985673529 7. William Stallings, Operating Systems: Internals and Design Principles, Pearson, 9th Edition, 2018, ISBN-10: 0134670957 • ISBN-13: 9780134670959 8. Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, Operating System Concepts, John Wiley & Sons, Inc., 10th Edition, 2018, ISBN-10: 1119439256 • ISBN-13: 978-1119439257		
Reference Books		
3. Tom Adelstein and Bill Lubanovic, Linux System Administration, O'Reilly Media, 1st Edition, 2007, ISBN-10: 0596009526 • ISBN-13: 978-0596009526 4. Harvey M. Deitel, Paul J. Deitel, and David R. Choffnes, Operating Systems, Prentice Hall, 3rd Edition, 2003, ISBN-10: 0131828274 • ISBN-13: 978-0131828278 5. Thomas W. Doeppner, Operating Systems in Depth: Design and Programming, Wiley, 1st Edition, 2010, ISBN-10: 0471687235 • ISBN-13: 978-0471687238 6. Andrew S. Tanenbaum and Herbert Bos, Modern Operating Systems, Pearson, 5th Edition, 2022, ISBN-13: 9780137618880 7. Milan Milenkovic, Operating Systems: Concepts and Design, 2nd Edition, McGraw-Hill Education, 2001, ISBN-10: 0074632728 • ISBN-13: 978-0074632727		
Online References		
1. NPTEL Course: Prof. Sorav Bansal, IIT Delhi, "Operating Systems" https://nptel.ac.in/courses/106102132 2. NPTEL Course: Prof. Santanu Chattopadhyay, IIT Kharagpur, "Operating System Fundamentals" https://nptel.ac.in/courses/106105214 3. NPTEL Course: Prof. Chester Rebeiro, IIT Madras, "Introduction to Operating Systems," https://nptel.ac.in/courses/106106144		

Second Year B.Tech Computer Engineering														
Semester- IV														
Course Code:CE24PCC253										Course Name: Software Engineering				
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Course Objectives: The Course aims : - To learn the fundamental concepts of software engineering, including software development methods, process models, and agile strategies. - To learn how to identify and analyze user needs and system requirements for developing well-defined software specifications. - To learn estimation techniques for analyzing software project effort and scheduling. - To learn how to model software systems structurally using Unified Modeling Language (UML) diagrams. - To learn the significance of Software Configuration Management (SCM) repositories and evaluate various software testing strategies.														
Course Outcomes: After learning the course, the students will be able to: CO1: Describe the concepts of software engineering methods, process models and agile Development Strategies. CO2: Apply user needs and system requirements analysis to develop comprehensive software requirements. CO3: Analyze project effort/schedule using appropriate estimation techniques. CO4: Draw Structural models of software systems using UML Diagrams. CO5: Recall the role of SCM repositories and the effectiveness of test strategies.														
Unit	Contents										Duration (Hrs.)			
1	Introduction to Software Engineering Nature of Software, Software Engineering Layers, Software Development Life Cycle (SDLC) Process models: Framework Activity, Process Pattern, waterfall model, incremental process models, evolutionary process models-Prototyping and Spiral, the unified process. Agile Software Development:Agility, Agile Process, Agile vs. Plan-Driven Development. Agile Methodologies: Introduction of Scrum, Extreme Programming (XP), Adaptive Software Development(ASD), Feature Driven Development(FDD), Introduction of Jira tool.										9			
2	Requirements Engineering Introduction to Requirements Engineering-Requirements Engineering, Establishing the groundwork,Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Requirements Negotiation and Validating Requirement, requirement analysis, scenario based modelling, requirement modelling strategies, flow oriented modelling, class based modelling, Software Requirements Specification (SRS).										9			

3	Estimation and Scheduling Software Project Estimation: Importance of Software Project Estimation, Estimation Decomposition Techniques-Software Sizing, Problem-Based Estimation, LOC-Based Estimation, FP-Based Estimation, The COCOMO II Model. Project Scheduling: Project Scheduling, Defining a Task for the Software Project, Scheduling.	9
4	Design Engineering and Risk Management Design Engineering-Design process, design concepts, the design model: Data Design, Architectural, Interface design elements. Architectural design-software architecture, architectural styles, architectural design and pattern, conceptual model of UML, basic structural modeling: class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams. Risk Management-Reactive vs Proactive risk strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.	9
5	Software Configuration Management and Testing Software Configuration Management: Software Configuration Management, The SCM Repository, The SCM Process. Version Control Systems: Git basics and collaboration. Testing Fundamentals: A Strategic Approach to Software Testing-Verification and Validation, Organizing for Software Testing, Software Testing Strategy—The Big Picture, Strategic Issues, Test Strategies for Conventional Software, Unit Testing, Integration Testing, Validation Testing, System Testing.	9
Total Hours		45
Text Books		
1. Roger S. Pressman's "Software Engineering: A Practitioner's Approach" 9th Edition, published in 2020 by McGraw Hill. 2. Dan Pilone and Neil Pitman's UML 2.0 in a Nutshell First Edition, published by O'Reilly Media in June 2005 3. Ian Sommerville's "Software Engineering" 10th edition, published by Pearson		
Reference Books		
4. Bob Hughes, Mike Cotterell, Rajib Mall: Software Project Management, 6th Edition, McGraw Hill Education, 2018. 5. An Integrated Approach to Software Engineering by Pankaj Jalote 3rd Edition, published by Springer in 2005. 6. "Software Engineering: Principles and Practice", Hans van Vliet, Wiley India, 3rd Edition, 2010. 7. Jim Arlow, Ila Neustadt, —UML 2 and the unified process –practical object-oriented analysis and design Addison Wesley, Second edition, 2005, ISBN 978-0201770605 8. Grady Booch et al, "Object-Oriented Analysis and Design with Applications", 3rd Edition, Pearson Education, 2007		
Online References		
1. NPTEL Course: Prof. Rajib Mall, IIT Kharagpur, "Software Engineering" (Link) 2. NPTEL Course: Prof. Ramesh Anbanandam, IIT Roorkee, "Project Management" (Link)		

Second Year B.Tech Computer Engineering														
Semester- IV														
Course Code: CE24PCC255					Course Name: Database Management Systems Laboratory									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	2	-	-	-	-	-	-	-	50	50	-	1	-	1
Prerequisite: CE24PCC252 - Database Management Systems														
Course Objectives: The course aims to: - Understand database concepts and ER/EER modeling. - Develop SQL and PL/SQL programs for database operations.. - Implement advanced SQL and PL/SQL features like joins, triggers, and cursors. - Explore NoSQL databases (CRUD, indexing, aggregation, MapReduce). - Integrate databases with front-end applications for real-world projects.														
Course Outcomes : After completing the course, the students will be able to: CO1: Apply database design concepts to create ER diagrams and relational models. CO2: Develop SQL queries using DML, DCL, and TCL commands. CO3: Implement joins, views, subqueries, procedures, and triggers in SQL. CO4: Design CRUD operations in MongoDB using NoSQL concepts.														
List of Experiments														
Sr. No.	Name of the Experiment												Duration (Hrs.)	
1	Case Study of ER diagram Study the ER (Entity-Relationship) diagram and design ER features using tools such as ERDPlus, ERwin, etc. Convert the ER diagram.												2	
2	SQL DDL Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym												2	
3	SQL DML Design at least 10 SQL queries for suitable database application using SQL DML statements: Insert, Select, Update, Delete with operators, functions, and set operators.												2	
4	JOINS Design at least 10 SQL queries for suitable database application using SQL DML statements:all types of Join, Sub-Query and View												2	
5	PL/SQL Stored Procedure and Stored Function.(use control structure and Exception handling) Write a Stored Procedure namely proc_Grade for the categorization of students. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks 899 and 825 category is Higher Second Class Write a PL/SQL block for using procedure created with above requirement												4	
6	Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor) Write a PL/SQL block of code using parameterized Cursor, that will merge the												4	

	data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped	
7	Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers). Write a database trigger on the Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in the Library_Audit table. Frame the problem statement for writing Database Triggers of all types, in-line with the above statement. The problem statement should clearly state the requirements.	4
8	MongoDB Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators)	2
9	Aggregation and indexing Implement aggregation and indexing with suitable examples using MongoDB.	2
10	Map reduce operation Implement Map reduce operation with suitable examples using MongoDB.	2
11	Database Connectivity: write a program to implement MySQL/Oracle database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)	4
Total Hours		30
Suggested Mini Projects List		
Note: 1. Design and implement a database application that showcases key DBMS features including ER modeling, SQL (DDL/DML), joins, subqueries, PL/SQL (procedures, functions, cursors, triggers), and front-end connectivity. Develop application considering 3. FrontEnd:Java 4. Backend:MongoDB/MySQL/Oracle		
P1	College Course Registration System Focus: ER Diagram, SQL DDL & DML, Normalization Design an ER diagram for student, course, department, and enrollment. Convert to relational tables, normalize to 3NF. Create tables using DDL; insert sample data using DML.	
P2	Online Library Management System Focus: SQL Views, Indexes, Triggers Create Book, Member, Issue, Return tables. Use views to show books currently issued. Create an index on Book Title. Create Before/After Triggers to log updates/deletes in a Library_Audit table.	
P3	Student Grading System Focus: Stored Procedures & Exception Handling Table: Stud_Marks(name, total_marks), Result(Roll, Name, Class) Create a procedure proc_Grade to assign classes based on marks. Call this procedure in a PL/SQL block with error handling.	
P4	Employee Payroll Management Focus: SQL DML, Joins, Subqueries Tables: Employee, Department, Payroll, Leaves Use inner, outer, self joins to display employee data. Write subqueries to find top-paid employees, unpaid leaves, etc.	

P5	Attendance Tracking System Focus: Parameterized Cursors Tables: N_RollCall, O_RollCall Use parameterized cursor to merge new roll calls into old, skip duplicates	6
P6	E-commerce Product Inventory Focus: MongoDB CRUD & Indexing Collections: products, orders, users Perform all CRUD operations using save, update, delete. Apply indexing on product category and price	
1.	Sales Data Analysis Focus: MongoDB Aggregation & MapReduce Collection: sales(date, item, qty, price) Use aggregation to calculate monthly sales per item. Use MapReduce to find best-selling product categories.	
P8	Hospital Patient Management Focus: Triggers & Stored Functions Tables: Patient, Treatment, Billing, Audit Create a stored function to calculate treatment cost. Triggers to store deleted/updated patient records in Audit.	
P9	Banking System Interface Focus: MySQL/Oracle DB Connectivity Build a GUI in Python/Java/PHP to interact with MySQL/Oracle. Perform database navigation: add, edit, delete customer/account data.	
P10	Movie Ticket Booking System (MongoDB) Focus: MongoDB + Frontend Connectivity Collections: movies, users, bookings Use a GUI (Node.js, React, or Python Tkinter) to connect to MongoDB. Allow ticket booking, cancellation, and user data update.	
Total Hours		30
Guidelines for Laboratory Conduction: - The instructor is expected to design assignments by considering the prerequisites, technological aspects, utility, and current trends related to the topic. The assignment policy should cater to average students while also including elements that challenge and engage advanced learners. The use of open-source software is encouraged. Based on the concepts covered, students should work in groups of 2-3 to implement a mini project.. - Operating System recommended: - 64-bit Open-source Linux or its derivative. - Programming tools recommended:MySQL, MongoDB etc.		
Guidelines for Students: - The laboratory assignments are to be submitted by students in the form of a journal. - Journal consists of Vision Mission of Institute & Department, certificate, Index table of contents and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept, Queries, sample input and expected output, conclusion).		
Guidelines for Laboratory Assessment: - Continuous assessment of laboratory work is done based on overall performance and Laboratory performance of students. - Each Laboratory assignment assessment should assign grade/marks based on rubrics with appropriate weightage. - Suggested rubrics for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficiency, punctuality and neatness.		
Text Books		

1. Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", McGraw Hill Publishers, ISBN 0-07-120413-X, 7th edition
2. Connolly T., Begg C., "Database Systems", 4th Edition, Pearson Education, 2002, ISBN 8178088614
3. D T Editorial Services "BIG DATA Black Book", Dreamtech Press ISBN 13 : 978935119931
4. Murach J., "Oracle SQL and PL/SQL", Murach Publishing, ISBN 978-1943872367, 3rd Edition.

Reference Books

1. C J Date, "An Introduction to Database Systems", Addison-Wesley, ISBN: 0201144719
2. S.K.Singh, "Database Systems: Concepts, Design and Application", Pearson Education, ISBN 978-81-317-6092-5
3. Kristina Chodorow, Michael Dierolf, "MongoDB: The Definitive Guide", O'Reilly Publications, ISBN: 978-1-449-34468-9
4. Adam Fowler, "NoSQL For Dummies", John Wiley & Sons, ISBN-1118905628
5. Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB Publications ISBN: 9788176569644, 9788176569644
6. Connolly T., Begg C., "Database Systems: A Practical Approach to Design, Implementation, and Management", Pearson Education, ISBN 978-1292061184, 6th Edition.

Online References

e-Books :

- Date C. J., "SQL and Relational Theory: How to Write Accurate SQL Code", O'Reilly Media, 2nd Edition.
- NPTEL Course: Prof. Sreenivasa Kumar, IIT Madras, "Introduction to Database Systems"
<https://nptel.ac.in/courses/106106220>
- NPTEL Course: Prof. Arnab Bhattacharya, IIT Kanpur, "Database Management Systems"
<https://nptel.ac.in/courses/106104135>
- MongoDB University Course: "MongoDB Basics (CRUD, Aggregation, Indexing)", MongoDB Inc. <https://www.mongodb.com/learn/mongodb-basics>

Second Year B.Tech Computer Engineering														
Semester- IV														
Course Code: CE24PCC261 Course Name: System Programming and Operating System Laboratory														
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	2	-	-	-	-	-	-	-	50	50	-	1	-	1
Prerequisite: CE24PCC207- Object Oriented Programming Laboratory														
Course Objectives: The course aims to: - Analyze the working of assemblers and macro processors. - Compare process scheduling algorithms using suitable schedulers. - Develop the ability to implement classical synchronization problems using semaphores and mutex. - Apply memory management techniques for efficient resource utilization.														
Course Outcomes : After completing the course, the students will be able to: CO1: Implement assembly and macro processing techniques to generate intermediate and machine code. CO2: Implement Process Scheduling algorithms using Schedulers. CO3: Implement Classical Synchronization Problems using semaphores and mutex. CO4: Compare page replacement policies for efficient memory management.														
Suggested List of Experiments (Any 6)														
Sr. No.	Name of the Experiment										Duration (Hrs.)			
1	Source Code Preprocessing- Pass 1 Assembler Simulate the first pass of an assembler for a simple assembly language program. The assembler should read the source code line by line, identify and process labels, symbols, and literals, maintain a Symbol Table and Literal Table, and handle assembler directives such as START, END, DC, and DS. The input is a sample assembly program containing symbols, directives, and instructions, and the output should include the Symbol Table, Literal Table, and intermediate code prepared for Pass 2 of the assembler.										4			
2	Machine Code Generation- Pass 2 Assembler Simulate the second pass of an assembler that converts intermediate code from Pass 1 into final machine code. The assembler should read the intermediate code, use the Symbol Table and Literal Table generated in Pass 1, and resolve all addresses for instructions, labels, and literals. It should process assembler directives like START, END, DC, and DS, replacing placeholders in the intermediate code with actual memory addresses.										4			
3	Macro Table Construction- Macro Processor Pass 1 Design and implement the first pass of a macro processor for a simple assembly language program. The program should read the source code line by line, identify all macro definitions, and construct a Macro Name Table (MNT) to store macro names, starting addresses in the Macro Definition Table (MDT), and the number of parameters, along with a Macro Definition Table (MDT) to										4			

	store the actual instructions of each macro with parameter placeholders. The input is a sample assembly program containing multiple macro definitions with or without parameters, and the output is the MNT and MDT ready for further processing by macro processor pass 2.	
4	Macro Expansion and Source Code Generation: Macro Processor Pass 2 Design and implement the second pass of a macro processor that expands all macro calls in a source assembly program using the MNT and MDT generated in Pass 1. The program should read the source code line by line, identify macro calls, replace them with the corresponding instructions from the MDT, and substitute macro parameters with the actual arguments provided in the call. The input includes the source program containing macro calls and the MNT and MDT from Pass 1, and the output is the fully expanded source code, along with the updated MNT and MDT for reference	2
5	Hospital Emergency Room- CPU Scheduling A hospital emergency room receives patients with varying illness severity, and each patient requires different treatment durations. Critical patients should be prioritized, but non-critical patients must also be attended in reasonable time. Each patient is represented as a process with arrival time, burst time (treatment time), and priority (severity: 1 = critical, 2 = moderate, 3 = mild). Implement two CPU scheduling algorithms: Preemptive Priority Scheduling, where a newly arrived patient with higher severity can interrupt the current treatment, and Non-Preemptive Priority Scheduling, where once treatment starts, it runs to completion before the next patient is scheduled. Calculate waiting time and turnaround time for each patient and display the treatment order using Gantt charts for both algorithms.	2
6	Call Center Customer Support- CPU Scheduling A customer support call center receives multiple incoming calls from clients simultaneously. Each call requires different handling times depending on the complexity of the query. To ensure fairness, no single caller should be kept waiting too long, and every call should get attention in a timely manner. Each call is represented as a process with an arrival time and burst time (handling duration). Implement Round Robin scheduling with a fixed time quantum to allocate the CPU to calls in a cyclic order. Calculate waiting time and turnaround time for each call and display the service order using a Gantt chart.	2
7	Online Order Processing- Producer-Consumer Problem Simulate an online food delivery system where multiple orders are placed by customers and processed by a limited number of chefs. Orders act as produced items, and chefs act as consumers. Use counting semaphores to control resource availability and synchronize producer and consumer threads, and add a mutex to ensure that only one thread accesses the shared buffer (critical section) at a time. The program should manage a shared buffer where producers add orders and consumers remove them, preventing the buffer from overflowing or running empty while maintaining correct order processing.	2
8	Library Database Access- Reader Writer Problem Simulate a library information system where multiple students can view book availability at the same time, but adding new books or updating records must be done exclusively. Implement Reader Priority, allowing concurrent access for readers while writers perform updates exclusively. Use proper synchronization to avoid race conditions, deadlocks, and starvation, ensuring reliable access to library data.	4

9	Bank Loan Resource Management - Banker's Algorithm Simulate a banking system where multiple customers request loans, and each loan requires allocation of limited resources such as cash, staff approval, and credit verification. Implement the Banker's Algorithm for deadlock avoidance to determine whether the system is in a safe state. The program should take as input the number of processes (loan requests), available resources, maximum demand for each process, and currently allocated resources. It should evaluate resource requests, allocate resources safely if possible, and identify unsafe states or potential deadlocks.	2
10	College Computer Lab - Page Replacement Algorithms Simulate a College computer lab where students open different programs on computers with limited memory. Each program requires memory pages, and when memory is full, some pages must be removed to load new programs. Implement page replacement algorithms such as FIFO, LRU, or Optimal to decide which pages to remove. The program should take as input a sequence of program requests and the memory size, then calculate the number of page faults for each algorithm.	4
Total Hours		30
Mini Projects List Note: 1. The Mini Project should be implemented in a group of 3-4 students. 2. Students can select any one from the following mini project titles. 3. Students can select a mini project title of their own choice.		
P1	Manufacturing System Design and develop a program that applies the Shortest Job First (SJF) scheduling algorithm using C/C++/Java to optimize task management in a manufacturing system. The system will manage a series of tasks in the production line, where each task requires different processing times (burst times). The goal is to prioritize shorter tasks, ensuring that they are completed before longer ones, thereby optimizing throughput and reducing overall waiting times in the production process.	6
P2	Customer Service Center Design and develop a system that applies the Round Robin scheduling algorithm using C/C++/Java to manage the allocation of customer queries to available agents in a customer service center for an e-commerce website. The system should ensure that each customer is served in turn, with an equal time slice given to each agent to handle incoming queries. The goal is to minimize customer wait times and ensure fairness in query handling, preventing any agent from being overwhelmed while others remain idle.	
P3	Multimedia Streaming Service (e.g. Netflix, YouTube) Design a solution to simulate the Producer-Consumer problem in a multimedia streaming service (e.g., Netflix, YouTube). In this system, the producer represents the servers that generate or stream video/audio data, while the consumer represents the devices (smartphones, laptops, etc.) that retrieve and play the media data. The program must manage a shared buffer that holds the streaming data, ensuring smooth playback by properly synchronizing the producer and consumer. The producer will generate video/audio data at a specific rate, while the consumer will process and display it. If the buffer is full, the producer must wait for space to be available; if the buffer is empty, the consumer must wait for data to be produced.	
P4	Collaborative Document Editing Systems (e.g. Google Docs) Design a solution to implement the following scenario, In a collaborative	

	document editing system, the goal is to enable multiple users to read the document simultaneously while ensuring that only one user can edit a specific section of the document at any given time. The system should implement a reader-priority synchronization approach, where reading operations are non-blocking unless a user is actively editing the document. The solution must ensure that multiple users can view the document concurrently without interference, while preventing conflicts during editing by granting exclusive access to the document when needed. The system must handle concurrent read and write operations efficiently, facilitating real-time collaboration while preventing race conditions and ensuring data consistency.	
P5	Printer Spooler System Design a solution to simulate a Printer Spooler System in an office network using synchronization techniques inspired by the Dining Philosophers Problem. The program must efficiently manage multiple users (threads) sending print jobs to a shared printer while avoiding deadlocks and ensuring fair access to the printer resource.	
P6	Airline Reservation Systems Implement a solution to manage seat availability in an airline reservation system, ensuring that flight bookings do not exceed the available seat capacity. The system must handle multiple customers attempting to book tickets simultaneously, and efficiently allocate seats without causing overbooking. The program should use the Banker's Algorithm to manage seat allocation, checking whether a customer's request can be safely granted without leading to an unsafe overbooking situation. This approach will ensure that the reservation process remains within capacity, avoiding resource contention and guaranteeing smooth operations, even when multiple booking systems or interfaces are interacting with the same flight data.	
P7	Search Engine Query Results Design and develop a system that applies the Least Recently Used (LRU) caching algorithm for a search engine system. The program should manage the cache of search query results, ensuring that the least recently used results are evicted when the cache reaches its limit, while the most frequently accessed results are retained. The system should efficiently update the cache as new queries are processed and old queries are no longer needed. This implementation should optimize the search process by reducing search times and enhancing the overall user experience by minimizing the need to recompute results for previously searched queries.	
P8	File System Design and implement a file system for an operating system that demonstrates both CPU and memory virtualization concepts. The file system should operate on a virtual disk and support core file system functionalities such as disk initialization, directory management, file creation, reading, writing, and deletion. The system must internally manage disk blocks, free space, and file metadata using inode- or FAT-based allocation techniques, without relying on the host operating system's file system services. Additionally, the project should simulate process execution and memory usage for file operations to illustrate how CPU time sharing and virtual memory mapping are applied during file access.	
Total Hours		30
Guidelines for Laboratory Conduction:		
1. Assignments on all concepts are mandatory.		

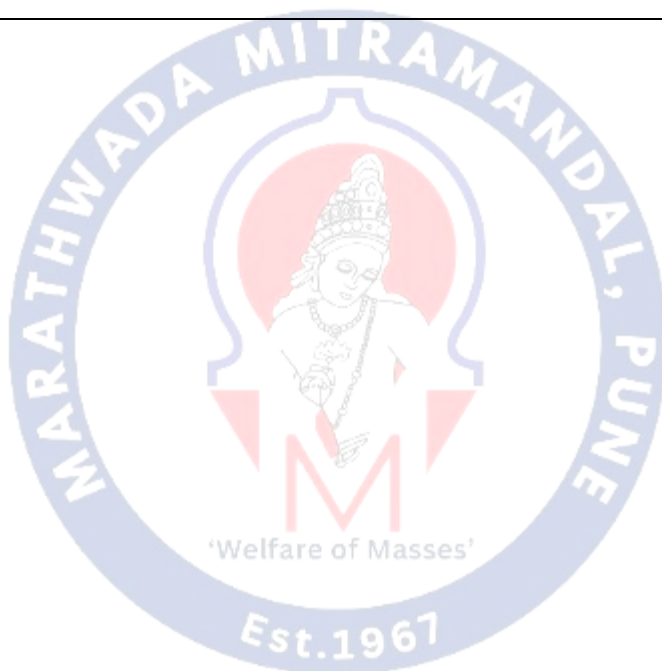
2. Assignments should be implemented on coding platforms such as GitHub, HackerRank, CodeChef.
3. Use of open-source software is encouraged.
4. Operating System recommended: - 64-bit Open-source Linux or its derivative.
5. Programming tools recommended: - G++/GCC, Eclipse.

Guidelines for Students:

1. The laboratory assignments are to be submitted by students in the form of a journal.
2. Journal consists of prologue, certificate, table of contents and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, Theory- Concept, algorithm , program code with suitable input and output, conclusion).

Guidelines for Laboratory Assessment:

- Continuous assessment of laboratory work is done based on overall performance and Laboratory performance of students.
- Each Laboratory assignment should be evaluated based on parameters with appropriate weightage.
- Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficiency, punctuality and neatness.



Second Year B.Tech Computer Engineering														
Semester- IV														
Course Code: CE24VSE257										Course Name: Web Technology				
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	4	-	-	-	-	-	50	-	-	50	-	2	-	2
Course Objectives: The course aims : - To gain a solid understanding of web essentials and markup languages. - To apply client-side technologies effectively in web development. - To utilize server-side technologies for building dynamic web applications. - To comprehend web services and frameworks for efficient web development.														
Course Outcomes: After learning the course, the students will be able to: CO1: Understand the core concepts of web development, including HTML, CSS,Bootstrap and their role in creating dynamic web pages. CO2: Apply JavaScript, and DOM to design responsive and interactive web pages that work across various devices. CO3: Apply jQuery for DOM manipulation, events, and effects. CO4: Analyze security risks in web applications, implementing best practices to secure applications from common vulnerabilities. CO5: Apply server side technology PHP to create Dynamic web pages														
Sr. No.	Name of the Experiment											Duration (Hrs.)		
1	Create a Webpage Using HTML: 5. Design a webpage that includes structured content like headings, paragraphs, lists, tables, forms, and images. Ensure semantic use of HTML tags to maintain accessibility and SEO best practices. 6. Provide a description of the layout and structure of the page.											4		
2	Style the Webpage Using CSS: 1. Apply various CSS properties to style the webpage. Include text formatting, background colors, borders, margins, and padding. 2. Implement the CSS Box Model to control element spacing and positioning.											4		
3	Enhance the Webpage Using JavaScript: 1. Add interactivity to your webpage by writing JavaScript code that can manipulate the DOM, such as changing element styles, displaying messages, or hiding/showing content based on user actions. 2. Implement basic event handling (e.g., button clicks, form submissions).											4		
4	Implement an Web Application using Bootstrap: To design a responsive web page using Bootstrap's grid system, components, and utilities. By the end of the lab, students will be familiar with key Bootstrap features and able to create a visually appealing and functional web layout.											4		
5	Integrate jQuery 4. Use jQuery to simplify DOM manipulation and enhance the user experience with animations and effects. 5. Handle events like clicks and hover to change content dynamically.											4		
6	XML Data Handling: 6. Create an XML document to store structured data (e.g., list of											4		

	products, users, or events) and process it using JavaScript or PHP. 7. Manipulate the XML document (e.g., read and modify elements) using DOM methods.	
7	Content Management System (CMS) Set Up WordPress CMS: - Explore the WordPress dashboard, create posts and pages, and install plugins. - Customize a WordPress theme by editing the CSS or creating a child theme.	4
8	Create a PHP-based Web Application: - Develop a PHP-based web application that accepts user input via an HTML form and processes it on the server side. For instance, create a contact form that sends the user's data to the server. - Integrate basic form validation to ensure secure data input.	4
Total Hours		32
Suggested Mini Projects List Note: - The Mini Project should be implemented in a group of 3-4 students. - Students can select any one from the following mini project titles. - Students can select a mini project title of their own choice.		
P1	Personal Portfolio Website Description: Create a personal portfolio website that showcases a student's skills, projects, and resume. It should include features like a contact form, an interactive design, and sections for skills, projects, and blogs.	28
P2	Online Resume Builder Description: Develop a web application that allows users to create, customize, and download their resumes in different formats (e.g., PDF, Word). It should offer pre-designed templates and allow users to input personal details, skills, and work experience.	
P3	Task Management Web App Description: Build a simple task management app where users can create, edit, delete, and mark tasks as completed. Add features like categorizing tasks, setting deadlines, and reminders.	
P4	Online Book Store Description: Create a simple online bookstore that allows users to browse books, view details, and add them to their cart. Include features like search, category filtering, and checkout (without actual payment gateway).	
P5	Weather Dashboard Description: Design a weather dashboard that allows users to check the weather of their location or any city by entering a city name. The app should display weather data such as temperature, humidity, wind speed, and weather conditions.	
P6	Recipe Sharing Platform Description: Develop a platform where users can share and search for recipes. It should allow users to post recipes with images, ingredients, and preparation steps, as well as rate and comment on others' recipes.	
P7	Online Quiz Application Description: Build a web-based quiz application where users can take	

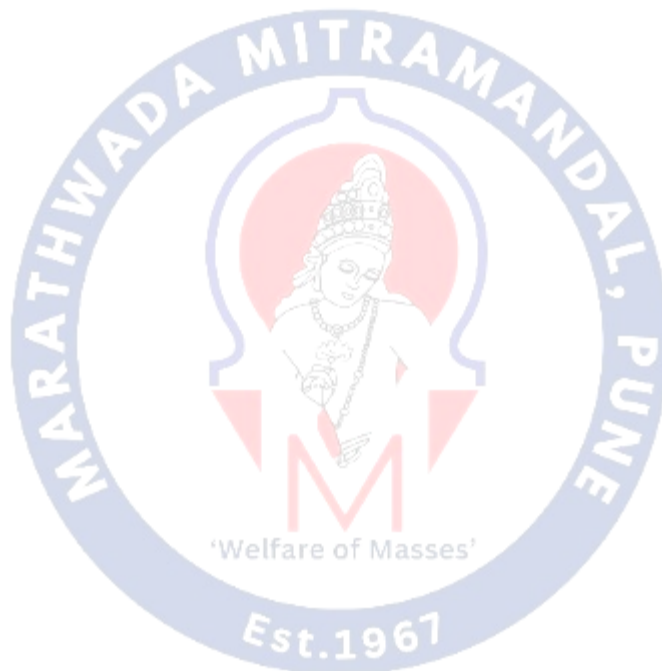
	quizzes on various topics. Features should include multiple choice questions, time limits, score tracking, and question randomization.	
P8	E-learning Platform Description: Create a basic e-learning platform that allows users to register, enroll in courses, and track their progress. Include features like quizzes, progress tracking, and course search.	
P9	Movie Review Website Description: Design a website where users can browse movies, read reviews, and submit their own reviews. Use an API to fetch movie data or build a small database for storing movie information.	
P10	Budget Tracker Description: Build a web application to track personal finances, where users can input income, expenses, and categorize their spending. Display charts and reports to show spending trends.	
Total Hours		60
Guidelines for Laboratory Conduction: - The instructor is expected to design assignments by considering the prerequisites, technological aspects, utility, and current trends related to the topic. The assignment policy should cater to average students while also including elements that challenge and engage advanced learners. The use of open-source software is encouraged. Based on the concepts covered, students should work in groups of 2-3 to implement a mini project.. - Operating System recommended: - 64-bit Open-source Linux or its derivative. - Programming tools recommended: - Eclipse/Geany, Apache/Tomcat server/PHP Guidelines for Students: - The laboratory assignments are to be submitted by students in the form of a journal. - Journal consists of Vision Mission of Institute & Department, certificate, Index table of contents and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, Date of Completion, Assessment grade/marks and assessor's sign, Theory-Concept, algorithm/pseudocode, sample input and expected output, conclusion). Guidelines for Laboratory Assessment: - Continuous assessment of laboratory work is done based on overall performance and Laboratory performance of students. - Each Laboratory assignment assessment should assign grade/marks based on rubrics with appropriate weightage. - Suggested rubrics for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficiency, punctuality and neatness.		
Text Books		
- Robert W. Sebesta, "Programming the World Wide Web", 4th Edition, Pearson education, 2008. - Web Technologies by Achyut Godbole and Atul Kahate, Publication: Tata McGraw-Hill Education Pvt. Ltd., ISBN 13: 9781259062681. - Web Technology: Theory and Practice by M. Srinivasan, Released June 2012, Publisher(s): Pearson India, ISBN: 9788131774199. "JavaScript & JQuery: The Missing Manual" David Sawyer McFarland, O'Reilly Media Edition: 2nd Edition (2014)		
Reference Books		
- Jon Duckett 'HTML and CSS: Design and Build Websites', Wiley, 2nd Edition, 2022. - Douglas Crockford, JavaScript: The Good Parts, O'Reilly Media, 1st Edition, 2008. - Bryan Sullivan and Vincent Liu, Web Security, McGraw-Hill Education, 1st Edition, 2011. - Erik T. Ray, Learning XML, O'Reilly Media, 4th Edition, 2019. - Steven Holzner, PHP and MySQL: The Complete Reference, McGraw-Hill Education, 1st		

Edition,2007

6. Web Technologies: A Computer Science Perspective, Jeffrey C.JacksonSecond Edition, Pearson Education, 2007, ISBN 978-0131856035.

Online References

4. <https://www.w3.org/html/>
5. HTML, The Complete Reference <http://www.htmlref.com/>
6. <http://w3schools.org/>
7. <http://php.net/>
8. <https://jquery.com/>
9. <https://developer.mozilla.org/en-US/docs/AJAX>
10. <http://www.tutorialspoint.com/css/>



Second Year B. Tech Computer Engineering															
Semester - III															
Course Code: CE24VEC259								Course Name: Universal Human Values							
Teaching Scheme (Hours/Week)					Examination Scheme						Credits				
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL	
-	-	2	-	-	-	-	50	-	-	50	-	-	2	2	
Prerequisites: • Readiness for Self-Exploration The course is not about memorizing facts but about a process of dialogue between what you are and what is naturally acceptable to you. You should be prepared to: - Verify proposals: Treat course content as "proposals" to be tested against your own Natural Acceptance rather than dogmas to be accepted blindly. - Validate experientially: Be willing to test these values in your daily living to see if they actually lead to harmony and happiness. • Openness to Holistic Thinking The syllabus moves from the individual to the family, society, and finally nature. You need an initial willingness to: - Look beyond individual success toward mutual fulfillment. - Acknowledge the interconnectedness between yourself and the "four orders" of nature (physical, bio, animal, and human).															
Course Objectives: • Development of a holistic perspective based on self- exploration about themselves (human-beings), family, society and nature/existence. • Understanding of the harmony in the human-beings,family, society and nature/existence. • Strengthening of self-reflection. • Development of commitment and courage to act.															
Course Outcomes: After the completion of course, the students can: CO1: Recognize the concept of self-exploration as the process of value education and see they have the potential to explore on their own right. CO2: Explore the human being as the coexistence of self and body to see their real needs / basic aspirations clearly. CO3: Explain relationship between one self and the other self as the essential part of relationship and harmony in the family CO4: Interpret the interconnectedness, harmony and mutual fulfilment inherent in the nature and the entire existence															
Unit	Content														
1	Introduction to Value Education and Harmony in the Human Being - Understanding Value Education - Self-exploration as the Process for Value Education														

	iii. Continuous Happiness and Prosperity- the Basic Human Aspirations and their Fulfilment iv. Right Understanding, Relationship and Physical Facility v. Happiness and Prosperity- Current Scenario vi. Method to Fulfil the Basic Human Aspirations	
2	Harmony in the Human Being : i. Understanding Human being as the Co-existence of the Self and the Body ii. Distinguishing between the Needs of the Self and the Body iii. The Body as an Instrument of the Self iv. Understanding Harmony in the Self v. Harmony of the Self with the Body vi. Programme to Ensure self-regulation and Health	
3	Harmony in the Family, Society and Nature : i. Harmony in the Family- the Basic Unit of Human Interaction "Trust"- the Foundational Value in Relationship ii. 'Respect'- as the Right Evaluation iii. Values in Human-to-Human Relationship iv. Understanding Harmony in the Society v. Vision for the Universal Human Order vi. Understanding Harmony in Nature – Interconnectedness and Mutual Fulfillment	
4	Professional Ethics and Application of Values: i..Competence in Professional Ethics: Ability to Utilize Values in Decision-Making ii. Ethical Human Conduct – The Foundation of Universal Human Order iii. Integration of Human Values with Technical Education	
Sr. No	List of Tutorial (Select any 08 activities out of 10 contributing to 26 hours)	Hours
1	Structured Self-Exploration Reflection Report	4
2	Case Study analysis on Contemporary Life Issues	4
3	Value Identification from Everyday Scenarios	4
4	Reflective Journal – Harmony in Self	2
5	Group Discussion - Topics e.g The Goal of Education (Siksha-Sanskar) ,Health and Responsibility (Swasthya-Sanyam), Production and Nature, The Cycle of Nature, Identifying the Root Cause, Defining Conduct	2
6	Role Play: Human Relationships	2
7	Debate: Material Growth versus Human Well-Being	2
8	Scenario Evaluation: Right Understanding in Personal Decisions	4
9	Engineering Ethics	2
10	Peer Interaction on Social Responsibility	2

Total Hours for 08 Activities		26
Text Books		
1. The Textbook A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034- 47-1 2. The Teacher"s Manual for A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G 3. P L Dhar, RR Gaur, 1990, Science and Humanism, Commonwealth Publishers. 4. E G Seebauer & Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers , Oxford University Press 5. AICTE Model Curriculum for Universal Human Values .		
Journals and Articles		
1. Dr. SARVESHWARI SHEKHAWAT (May-2025). Universal Human Values and Professional Ethics: A Path to Sustainable Development. INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT, 10(5), 276-287. https://ijnrd.org/papers/IJNRDD001043.pdf 2. Rani, S., Beenu, Jindal, H., Gautam, N., & Kumar, S. (2022). Importance of Universal Human Values for Life: A Study. Asian Journal of Science and Applied Technology, 11(1), 36-48. https://doi.org/10.51983/ajsat-2022.11.1.3204		
Online References		
• Exploring Human Values: Visions of Happiness and Perfect Society (https://nptel.ac.in/courses/109104068) • Moral Thinking: An Introduction to Values and Ethics (https://onlinecourses.nptel.ac.in/noc24_hs169/preview) • NPTEL Course: "Human Values and Ethics" by Prof. V. S. Ravi Link: NPTEL - Human Values and Ethics • NPTEL Course: "Human Values" by Prof. R.R. Gaur (IIT Kanpur) Link: NPTEL - Human Values • Value Education Workshop Videos by AICTE/UGC Link: https://www.aiCTE-india.org > Education > Universal Human Values • UGC Video Resources for Human Values Search: "UGC Human Values Video Lectures" on YouTube or SWAYAM		

Second Year B.Tech Computer Engineering														
Semester-IV														
Course Code: SH24AEC202					Course Name: Professional Communication and Personality Development									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	-	2	-	-	-	-	50	-	-	50	-	-	2	2
Prerequisite: Basic understanding of LSRW skills, grammar, and vocabulary of the English Language.														
Course Objectives: - To develop students' rhetorical and persuasive abilities for effective oral communication and audience engagement. - To familiarize students with key business and managerial terminologies essential for effective communication and decision-making. - To strengthen analytical and critical thinking skills for constructing logical arguments and making informed decisions. - To cultivate professional and technical writing competence, ensuring clarity, accuracy, and etiquette in communication.														
Course Outcomes: After learning the course, the students will be able to: CO1: Demonstrate effective rhetoric, persuasive skills, audience engagement, and adaptability in communication. CO2: Apply visual aids, body language, voice modulation, and confidently manage audience questions and maintain engagement throughout. CO3: Analyze information and construct logical arguments to solve problems and make informed Decisions. CO4: Exhibit professional and technical writing skills and avoid common pitfalls, ensuring clear, polite, and informative communication.														
Contents														
1	Public Speaking Using Persuasive language and understanding the context in public speaking Controlling nervousness, building confidence, stage presence, poise, and impact Employing voice modulation, pace, volume & pitch.													
2	Presentation Skills Structuring a presentation & using visual aids Understanding audience connection strategies & feedback Handling Q&A sessions with confidence.													
3	Critical Thinking Working on Case Studies and Situational Analysis Structuring the argument and handling rebuttal Analyzing logical fallacies and cognitive biases.													
4	Technical Writing Structuring emails for different purposes (requests, complaints, updates, and follow-ups) Avoiding common mistakes and making the right use of CC, BCC, and reply-all options Blog/article/ newsletter/survey report format and examples.													
5	Group Discussion													

	Implementing coherence and cohesion in structuring the argument Integrating non-verbal communication in the conversation Managing disagreement and aggression in the discussion.	
List of Tutorials		
Sr. No.	Name of the Tutorial	Duration (Hrs.)
1	Ice-breaking Session	3
2	Storytelling	4
3	Idea Presentation	4
4	Advertisement Creation	4
5	Case Study Analysis	2
6	Debate	3
7	Email Writing (Professional & Personal)	2
8	Blog Writing	2
9	Survey Reports	2
10	Group Discussion	4
Total Hours		30
Text Books		
1. Stephen E. Lucas, <i>The Art of Public Speaking</i> , McGraw Hill Publication, ISBN-10: 1260914275 ISBN-13: 9781260914276. 2. Jaishri Jethwaney and Sanjay Bhargava, <i>Corporate Communication: Principles and Practices</i> . 3. Dr. Ranjit Singh, <i>Critical Thinking and Problem Solving</i> . 4. Rajendra Pal and J. S. Korlahalli, <i>Essentials of Business Communication</i> , Sultan Chand & Sons, 2011. ISBN: 8180547299, 9788180547294.		
Reference Books		
1. Shiv Khera, <i>You Can Win</i> , Bloomsbury India, ISBN-10: 9382951717 · ISBN-13: 9789832951711. 2. Carmine Gallo, <i>Talk Like TED</i> , Macmillan Business. 3. Garr Reynolds, <i>Presentation Zen: Simple Ideas on Presentation Design and Delivery</i> , New Riders. 4. Peter Facione and Carol Ann Gittens, <i>Think Critically</i> . 5. Lewis Vaughn, <i>The Power of Critical Thinking</i> , Oxford University Press. 6. Emily Post and Peter Post, <i>The Etiquette Advantage in Business: Personal Skills for Professional Success</i> .		
Online References		
1. NPTEL Course: <i>Technical English for Engineers</i> – https://onlinecourses.nptel.ac.in/noc20_hs56/preview 2. SWAYAM Course: <i>English for Research Paper Writing</i> – https://onlinecourses.swayam2.ac.in/ntr24_ed15/previe 3. SWAYAM Course: <i>Personality Development and Communication Skills</i> – https://onlinecourses.swayam2.ac.in/cec22_cm03/preview		