



Dnyaan Prasad Global University
Dr. D. Y. Patil Unitech Society
School of Technology and Research

Batch: 2025-2029 (CSE- Reg) / 2025-2028 (CSE- Lat)

SEMESTER 3

Catalog Course Code	Course Code	Course Type	Course Title	Engagement Hrs/Week				Credits	Continuous Assessment		Term End Examination		Total
				(Teaching Scheme)					Theory	Practical	Theory	Practical	
				L	T	P	Total						
STENCS019V0	DP04020425301	PCC	Data Structures and Algorithms	3	-	-	3	3	45	0	30	-	75
STENCS020V0	DP04020425302	PCC	Coding Café -OOP	2	-	-	2	2	30	0	20	0	50
STENCS021V0	DP04020425303	PCC	Computer Organization & Architecture	3	-	-	3	3	45	0	30	0	75
Multidisciplinary Minor Course (Choose Anyone)													
1. Minor in IoT and Embedded Systems			2. Minor in Hybrid Vehicles			3. Minor in Mechanical Design & CAD			4. Minor in Robotics and Automation				
STENCS106V0	DP04020325304	MDM	Embedded System Fundamentals ¹	3	-	-	3	3	45	0	30	0	75
STENCS211V0	DP04020325305	MDM	Fundamentals of Hybrid and Electric Vehicles ²	3	-	-	3	3	45	0	30	0	75
STENCS212V0	DP04020325306	MDM	Fundamentals of Computer-Aided Design ³	3	-	-	3	3	45	0	30	0	75
STENCS213V0	DP04020325307	MDM	Principles of Robotics ⁴	3	-	-	3	3	45	0	30	0	75

Open Elective-I (Choose Anyone)															
STENCS108V0	DP04020325308	OC-I	Digital Literacy & Cyber Awareness	3	-	-	3	3	45	0	30	0	75		
STENCS214V0	DP04020325309	OC-I	Probability and Statistics for AI Applications	3	-	-	3	3	45	0	30	0	75		
STENCS022V0	DP04020325310	E/E/M	Entrepreneurship & Startup Studio	2	-	-	2	2	-	50	0	0	50		
STENCS215V0	DP04020325311	VEC	Universal Human Values and Ethics	2	-	-	2	2	50	0	0	0	50		
STENCS024V0	DP04020425308	PCC	Data Structures and Algorithms Lab	-	-	2	2	1	0	0	0	25	25		
Multidisciplinary Minor Course Lab (Choose Anyone)															
1. Minor in IoT and Embedded Systems				2. Minor in Hybrid Vehicles				3. Minor in Mechanical Design & CAD				4. Minor in Robotics and Automation			
STENCS107V0	DP04020325313	MDM	Embedded System Fundamentals Lab ¹	-	-	2	2	1	0	25	0	0	25		
STENCS216V0	DP04020325314	MDM	Fundamentals of Hybrid and Electric Vehicles Lab ²	-	-	2	2	1	0	25	0	0	25		
STENCS217V0	DP04020325315	MDM	Fundamentals of Computer-Aided Design Lab ³	-	-	2	2	1	0	25	0	0	25		
STENCS218V0	DP04020325316	MDM	Principles of Robotics ⁴	-	-	2	2	1	0	25	0	0	25		
DPFPST02	DP04020425310	CEP /FP	Mini Project	-	-	4	4	2	0	0	0	50	50		
STPHGP002V0	DP04020425311	MNC	Environmental Studies	2	-	-	2	-	0	0	0	0	0		
Total Academic Engagement and Credits				20	-	8	28	22	210	125	140	75	550		

Program	B.Tech. (Computer Science & Engineering)			Semester		III		
PCC	Data Structures and Algorithms			Code		DP04020425301		
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
3	3	--	--	45	--	30	--	75
Prerequisite: Problem Solving through Programming								
Other Courses in this Series: Data Structures and Algorithms Lab								
Course Objectives: <ol style="list-style-type: none"> 1. To develop a strong foundation in algorithm analysis 2. To understand the role of abstract data types (ADTs) in software design and modular programming. 3. To learn core data structures such as arrays, linked lists, stacks, queues, trees, and graphs. 4. To implement and analyze searching, sorting, and hashing techniques for efficient data handling 5. To identify and use the most suitable data structures and algorithms for practical implementation 								
Course Outcomes: On completion of the course, students will be able to: <p>CO1: Analyze various data structures and algorithmic strategies to evaluate their efficiency in terms of time and space complexity.</p> <p>CO2: Design optimized searching, sorting, and traversal techniques for structured data organization and retrieval.</p> <p>CO3: Apply appropriate linear and non-linear data structures such as stacks, queues, linked lists, trees, and graphs to solve real-world problems.</p> <p>CO4: Create efficient solutions for dynamic data management, including memory allocation, indexing, and data representations.</p> <p>CO5: Develop algorithms for hierarchical data management, dynamic memory handling, indexing.</p> <p>CO6: Implement solutions to real world problems using advanced data structures like Graphs and Optimal binary trees.</p>								
Course Contents								
Unit I	Introduction to Data Structures and Algorithm Analysis						6 Hours	
Introduction to data, information, knowledge, and data structures. Abstract Data Types (ADT) and classification of data structures: linear and non-linear, static and dynamic, persistent and ephemeral. Basics of algorithms, problem-solving approaches, algorithm representation using pseudo-code and flowcharts. Algorithm complexity analysis: time and space complexity, step count method, and asymptotic notations (Big-O, Theta (Θ), Omega (Ω)). Complexity analysis of common programming constructs. Concept of sequential organization and arrays as an Abstract Data Type (ADT), Storage representation using row-major and column-major form, Multidimensional Array, Ordered List, Polynomial representation using Array.								
Case Study#1:	Consider any data processing department in an organization and identify the different data used, classify the data used, list the techniques used in data processing and analyse time complexity of the time and space complexity.							
Unit II	Searching and Sorting						6 Hours	

Need for Searching and Sorting, Classification of Sorting as Internal and External Sorting, and Concept of Sort stability. Linear Search, Sentinel Search, Binary Search, Bubble Sort, Insertion Sort, Selection Sort, Heap: Heap data structure, Min and Max Heap, Heap sort, applications of heap. Comparison of Sorting Algorithms based on Efficiency, Stability, and number of passes. Time and Space Complexity Analysis for Searching and Sorting methods in Best, Worst, and Average cases. Introduction to Hashing and its role in efficient searching. Different types of Hashing with its Applications.		
Case Study#2:	A retail store maintains an extensive inventory of products, including item names, prices, stock quantities, and supplier details. As the number of products increases, the store faces challenges in efficiently searching for items, managing stock levels, and organizing products for quick access. Design a solution by selecting appropriate searching and sorting techniques to improve inventory management and retrieval efficiency. Write time and space complexities for the solution designed.	
Unit III	Stack and queue	6 Hours
Sparse matrix representation, addition, and transposition using simple and fast transpose techniques. Introduction to Stacks and Queue as ADTs, representation using Array. Stack Applications in Recursion, Infix-to-Postfix and Infix-to-Prefix Conversion, and Postfix/Prefix evaluation. Types of Queue and Applications of Queue.		
Case Study#3:	A text editor application needs to implement an Undo-Redo feature that allows users to revert or reapply changes efficiently. The challenge is to track user actions (such as typing, deleting, or formatting) and enable seamless navigation between previous and later states of the document. Design a solution for handling this issue.	
Case Study#4:	A movie theatre has a ticket booking system where customers are served in a first-come, first-served manner. The system should efficiently manage the queue of customers waiting for tickets, handle cancellations, and prioritize VIP customers when needed.	
Unit IV	Linked List	6 Hours
Introduction to Static and Dynamic Memory Allocation and the Concept of Linked Organization. Linked List realization using Dynamic Memory Management. Types of Linked List: Singly linked list, Circular Linked List, doubly linked list, and Circular Doubly Linked List. Singly Linked List as an Abstract Data Type (ADT) with operations as Creation, Traversal, Search, Insertion, Deletion, Sorting, and Concatenation, Applications of linked lists. Merging of two sorted SLLs, Polynomial manipulations using Singly linked list. Introduction to Generalized Linked List (GLL) and its applications.		
Case Study#5:	A University maintains student records, including student ID, name, course, and grades . As students enrol, drop courses, or update their records, a flexible and efficient data structure is needed to handle dynamic changes. Design a solution using linked list the system should allow insertion, deletion, searching, sorting, and updating student records dynamically without memory wastage	
Case Study#6:	Study how operating systems manage dynamic allocation of main memory.	
Unit V	Tree	6 Hours
Introduction to Tree, Tree Terminologies. Binary Tree as ADT, Expression Tree. Binary Search Tree (BST) and Operations: Insertion, Search, Deletion, and Level-wise Display. Recursive and non-Recursive Tree Traversal Algorithms: Inorder and Preorder and Postorder, Introduction to Threaded Binary Tree, Insertion and deletion of nodes, Threaded Binary Search Tree Traversal techniques for in-order threaded binary tree, AVL Tree. Applications of Tree. Indexing and Multiway Trees: Introduction to B and B+ Tree.		

Case Study#7:	A library management system needs an efficient way to store, search, and retrieve book records based on various attributes like ISBN, title, and author . Additionally, the system should be able to evaluate complex search queries , requiring efficient expression evaluation. Design a solution using binary trees.	
Unit VI	Graph and Symbol Table	6 Hours
Introduction to graphs, Graph terminologies, Graph as an Abstract Data Type (ADT). Representation of graphs using Adjacency Matrix, Adjacency List. Graph Traversal Techniques: Breadth-First Search (BFS) and Depth-First Search (DFS). Construction of Minimum Spanning Trees (MSTs) using Prim's and Kruskal's Algorithms. Shortest Path using Dijkstra's algorithm and Topological Sorting. Symbol Table- Representation of Symbol Tables- Static tree table and Dynamic tree table		
Case Study#8:	The city transportation department needs to design an optimal road network and provide the shortest routes between important locations. Additionally, a database indexing system is required to efficiently store and retrieve city locations and road details. Design a solution for the above problem.	
Total Hours: 36		
Textbooks:		
<ol style="list-style-type: none"> Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, <i>Fundamentals of Data Structures in C</i>, 2nd Edition, Universities Press, 2008. ISBN: 978-8173716058 Aaron M. Tenenbaum, Yedidyah Langsam, and Moshe J. Augenstein, <i>Data Structures Using C and C++</i>, 2nd Edition, Pearson Education, 2009. ISBN: 978-0131997462 Mark Allen Weiss, <i>Data Structures and Algorithm Analysis in C</i>, 2nd Edition, Pearson Education, 2006. ISBN: 978-0321441461 		
Reference Books:		
<ol style="list-style-type: none"> Seymour Lipschutz, <i>Data Structures</i>, Schaum's Outlines Series, McGraw Hill, 2014. ISBN: 978-0070601681 Debasis Samanta, <i>Classic Data Structures</i>, 2nd Edition, PHI Learning, 2009. ISBN: 978-8120337315 Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, <i>Data Structures and Algorithms</i>, 1st Edition, Pearson Education, 1983. ISBN: 978-0201000238 Y. Kanetkar, <i>Data Structures Through C++</i>, 4th Edition, BPB Publications, 2023. ISBN : 978-9355511898 		
E-Books/Online Learning Resources:		
<ol style="list-style-type: none"> Robert Lafore, <i>Data Structures and Algorithms in Java</i>, 2nd Edition, Sams Publishing, 2002. ISBN: 978-0672324536 https://www.amazon.in/Data-Structures-Algorithms-Robert-Lafore/dp/0672324539 George Heineman, Gary Pollice, Stanley Selkow, <i>Algorithms in a Nutshell</i>, 2nd Edition, O'Reilly Media, 2016. ISBN: 978-1491948927 https://www.oreilly.com/library/view/algorithms-in-a/9781491912973/ 		
Online Courses:		

1. Problem Solving through Programming in C

- a. Instructor: Prof. Anupam Basu (IIT Kharagpur)
- b. Covers C programming basics and problem-solving techniques.
- c. Link: https://onlinecourses.nptel.ac.in/noc22_cs101/preview

2. Crash Course on Data Structure and Algorithms in C++

- a. Instructor: Great Learning Academy Team
- b. Covers fundamental data structures (arrays, linked lists, trees) and algorithmic problem-solving using C++.
- c. Link: <https://www.mygreatlearning.com/academy/learn-for-free/courses/crash-course-on-data-structure-and-algorithms-in-c>

CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	1	2	1	1	1	1	1	1	1
CO2	3	3	3	2	1	1	1	1	1	1	1
CO3	2	3	3	2	1	1	1	2	1	1	1
CO4	3	3	3	2	2	1	1	1	1	1	1
CO5	3	3	2	2	1	1	1	1	1	1	1
CO6	2	2	3	3	1	1	1	1	1	1	2

Program	B.Tech. (Computer Science & Engineering)			Semester	III			
PCC	Coding Café –Object Oriented Programming (OOP)			Code	DP04020425302			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		
				Theory	Practical	Theory	Practical	Total
2	2	--	--	30	--	20	--	50
Prerequisite:								
<ul style="list-style-type: none"> 1) Problem solving through programming 2) Computational thinking and programming. 								
Other Courses in this Series:								
<ul style="list-style-type: none"> 1) Mini Project 								
Course Objectives:								
<ul style="list-style-type: none"> 1. To introduce Object-Oriented Programming (OOP) concepts and their applications. 2. To develop problem-solving skills using OOP principles such as encapsulation, inheritance, polymorphism, and abstraction. 3. To enhance understanding of modular and reusable code design. 4. To apply OOP concepts in real-world software development. 5. To integrate exception handling, file handling, and collections in programming. 6. To develop a hands-on project implementing all OOP principles. 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Apply fundamental OOP concepts such as classes, objects, encapsulation, and abstraction.								
CO2: Implement modular programs using inheritance and polymorphism.								
CO3: Design real-world applications using OOP principles.								
CO4: Use exception handling techniques for robust programming.								
CO5: Demonstrates efficient use of STL containers and sorting algorithms.								
Course Contents								
Unit 1	Introduction to OOP						6 Hours	
Introduction- Need for Object-Oriented Programming, Difference between Procedural and Object-Oriented Programming, Classes, Objects, Attributes, and Methods, Static vs Dynamic Binding, Introduction to an Object-Oriented Language.								
C++ Programming- C++ programming Basics, Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, Access specifiers, separating interface from implementation.								
Functions- Function, function prototype, accessing function, Constructors and destructor, Types of constructors, Objects and Memory requirements, Static members: variable and functions, inline function, friend function, friend class.								
Case	Examine how AutoCAD models geometrical objects like lines, arcs, and shapes as C++							

Study#1:	classes and utilizes OOP concepts such as encapsulation, composition, and abstraction for design features.	
Case Study#2:	Learn how real-world banking systems are modeled using classes and objects.	
Unit 2	Encapsulation and Pointers	6 Hours
Understanding Encapsulation and Access Modifiers, Properties with controlled access through getter and setter Methods, Concept of mutable and immutable objects. Pointers: declaring and initializing pointers, indirection Operators, Memory Management: new and delete, Pointers to Objects, this pointer, Pointers Vs Arrays, accessing Arrays using pointers, Arrays of Pointers, Function pointers, Pointers to Pointers, Pointers to Derived classes, passing pointers to functions, Return pointers from functions, Call by value ,call by reference, Null pointer, void pointer.		
Case Study#3:	Design a simple Student Record System where student details such as roll number, name, and marks are stored privately. The data should only be accessed or modified using public functions. Additionally, manage student objects using pointers to demonstrate dynamic memory access and manipulation.	
Case Study#4:	Design a Bank Account Management System using C++ that securely stores customer details like account number, name, and balance using encapsulation. The program should allow users to create accounts dynamically using pointers and access data using pointer-based syntax	
Unit 3	Inheritance and Polymorphism	6 Hours
Inheritance: Introduction to Inheritance, Types of Inheritance (Single, Multiple, Multilevel, Hierarchical, Hybrid), Syntax and Basic Implementation, Constructors and Destructors in Inheritance, Access Control and Inheritance, Function Overriding, Use of Scope Resolution Operator, Virtual Base Classes, Abstract Classes, this keyword Polymorphism: Introduction to Polymorphism, Types of Polymorphism, Compile-time Polymorphism - Function Overloading, Operator Overloading, Run-time Polymorphism-Virtual Functions, Base class pointer/reference calling derived class functions, Late binding and dynamic dispatch, Abstract Classes and Pure Virtual Functions, Virtual Table (vtable) and Virtual Pointer (vptr).		
Case Study#5:	A company needs to manage different types of vehicles – Car, Truck, and Motorbike – with some common properties and specific behaviour. Identify how inheritance is used to avoid redundancy. Study how polymorphism helps in runtime decision-making using Vehicle* v. Give examples of how virtual functions are used and when to apply override.	
Case Study#6:	A company wants to store employee records (ID, name, basic salary, allowances) and generate pay slips based on this data, all stored and processed from a file. Design a class Employee with methods to input and display details.	
Unit 4	Exception Handling and File Handling	6 Hours
Exception Handling: Try, Catch, Finally, Throw, Throws , Exception Handling Mechanism, Re-throwing Exceptions, Exception Specification, Nested try-catch Blocks, Standard Exception Classes, Error Handling and Best Practices. File Handling: File Stream Classes, Opening and Closing Files, Reading from and Writing to Files, File modes, File Pointers and Their Manipulation, Detecting End of File, Binary File Handling, Error Handling		

in File Operations.	
Case Study#7:	A calculator program performs division. If the denominator is zero, it should throw an exception and handle it gracefully without crashing. Identify the role of throw, try, and catch.
Case Study#8:	An inventory system maintains a list of products (product ID, name, quantity, and price). The data is stored in a file. Users can add, update, or delete products, and the file must reflect the current inventory. Implement add, update, and delete functions on inventory records stored in a file.
Unit 5	Basic Template and Standard Template Library 6 Hours
Introduction to Template, Class Template, Function template, Template with Multiple Parameters, Generic Template, Non-Type Template Parameters STL Containers, Iterators, Algorithms, Stacks, Queues, and Priority Queues, Exception Safety and Performance	
Case Study#9:	Design a generic stack class that works for different data types using class templates.
Case Study#10:	Analyze the advantages and limitations of sequential and random-access file organizations. Provide examples of where each is best suited in real-world systems.
Total Hours: 30	
Textbooks:	
<ol style="list-style-type: none"> 1. Robert Lafore, "Object-Oriented Programming in C++", 4th Edition, Sams Publishing, ISBN: 0672323087, ISBN-13: 9780672323089 2. E. Balagurusamy, "Object-Oriented Programming with C++", 7th Edition, McGraw Hill Education, ISBN: 9353162343, ISBN-13: 9789353162349 3. Yashavant Kanetkar, "Let Us C++", BPB Publications, ISBN: 9387284495, ISBN-13: 9789387284495 	
Reference Books:	
<ol style="list-style-type: none"> 1. Bjarne Stroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley, ISBN: 0321563840, ISBN-13: 9780321563842 2. Bjarne Stroustrup, "Programming: Principles and Practice Using C++", 2nd Edition, Addison-Wesley, ISBN: 0321992784, ISBN-13: 9780321992789 3. Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, "C++ Primer", 5th Edition, Addison-Wesley, ISBN: 0321714113, ISBN-13: 9780321714114 	
E-Books/Online Learning Resources:	
<ol style="list-style-type: none"> 1. C++ Programming – Wiki books URL: https://en.wikibooks.org/wiki/C%2B%2B_Programming 2. Thinking in C++ URL: https://mindview.net/Books/TICPP/ThinkingInCPP2e.html 3. Learn C++ – LearnCpp.co URL: https://www.learncpp.com 4. C++ Notes for Professionals URL: https://goalkicker.com/CPlusPlusBook/ 	
Online Courses:	

1. Coursera – C++ For C Programmers
 - a. Instructor: Ira Pohl
 - b. Cover: Coursera – C++ For C Programmers
 - c. Link: <https://www.coursera.org/learn/c-plus-plus-a>
2. Introduction to Programming through C++
 - a. Instructor: Prof. Abhiram Ranade
 - b. Cover: Introduction to Programming through C++
 - c. Link: <https://youtu.be/efXI8anQwXo?si=bpx3nIFFU8TeVg8o>
3. Udemy – Beginning C++ Programming - From Beginner to Beyond
 - a. Instructor: Frank Mitropoulos
 - b. Cover: Udemy – Beginning C++ Programming - From Beginner to Beyond
 - c. Link: <https://youtu.be/l3hkjureXoY?si=2QpXWjRzGNPwIxov>
4. GeeksforGeeks – C++ Programming Language
 - a. Instructor: Parikshit Kumar Pruthi
 - b. Cover: GeeksforGeeks – C++ Programming Language
 - c. Link: <https://youtube.com/playlist?list=PLqM7aIHxFySGg6GSRmE2INI4k8fPH5qVB&si=Xtl7nVI-w8sQrrif>

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	-	2	-	-	-	-	-	1
CO2	3	2	3	-	2	-	-	1	-	-	1
CO3	2	3	3	1	2	1	-	1	1	1	1
CO4	2	3	2	1	3	-	-	-	-	-	1
CO5	3	2	2	-	3	-	-	-	-	-	2

Program	B.Tech. (Computer Science & Engineering)			Semester		III		
Course	Computer Organization & Architecture			Code		DP04020425303		
Teaching Scheme (Hrs/Week)				Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theor y	Practical	Theory	Practical	
3	3	--	--	45	--	30	--	75
Prerequisite:								
1) Computer Essentials Workshop 2) Digital Logic Design & Microprocessor								
Other Courses in this Series:								
1) NA								
Course Objectives:								
1. Conceptualize the basics of organizational and architectural issues of a digital computer. 2. Comprehend the principles of data representation, arithmetic operations, and their hardware implementation. 3. Learn about the design of processors, memory hierarchies, and instruction execution. 4. Study processor organization and parameters influencing performance of a processor. 5. To understand various data transfer techniques in digital computers. 6. Learn microprocessor architecture and study assembly language programming.								
Course Outcomes: On completion of the course, students will be able to:								
CO1: Understand the basic architecture and components of a computer. CO2: Analyze and evaluate computer arithmetic. CO3: Elaborate instruction set architecture and its impact on hardware design. CO4: Demonstrate ALU & Control unit design in modern processors. CO5: Categorize the different levels of memory organization and explain its function. CO6: Analyze I/O organization; interrupt handling, and direct memory access in the context of contemporary systems.								
Course Contents								
Unit 1	Overview of Computer Organization and Architecture						6 Hours	
Introduction of Computer Organization and Architecture, Evolution of Computers, Von Neumann model Basic organization of computer and block level description of the functional units (Basic Operational Concepts, Bus structures, Memory Locations, Addressing Mode (Immediate, Register, Direct, Indirect, Indexed, PC-Relative, Auto-Increment/Decrement)). Performance measure of Computer Architecture								
Case Study#1:	Basic Architecture of 8086 microprocessor							
Case Study#2:	Addressing modes & Register Organization of 8086							
Unit 2	Data Representation and Computer Arithmetic						6 Hours	

Floating- Point Representation (IEEE 754 Standard: single and double precision). Floating-Point Arithmetic: Addition, Subtraction, Multiplication, Division. Scalar Data Types, Fixed-Point Arithmetic: Addition, Subtraction, multiplication of positive numbers, signed operand multiplication, booths algorithm, fast multiplication, integer Division (Restoring/Non-Restoring)		
Case Study#3:	Study Booth's Algorithm and Division methods.	
Case Study#4:	Study Floating-Point in Graphics (IEEE 754) Compliance for NVIDIA GPUs.	
Unit 3	Instruction Pipelining and CPU Structure	7Hours
CPU Architecture, Instruction formats, basic instruction cycle, Instruction set, Instruction Pipelining: Pipeline processing, instruction pipelining, pipeline stages (Case Study- Intel 586 pipelining.), Pipeline Hazards: Structural, Data (RAW, WAR, WAW), Control (Branch prediction). Hazard Resolution Techniques: Stalling, Forwarding (Bypassing), Branch Prediction (Static/Dynamic, BTB, Predictors). Instruction level and machine parallelism, Instruction issue policy, register renaming, machine parallelism, branch prediction, superscalar execution and implementation.		
Case Study#5:	Instruction Set of 8086 microprocessors	
Case Study#6:	Study the examples on instruction pipelining of Intel 586 Pipelining	
Unit 4	Register organization, ALU design and Control unit design	6 Hours
Register transfer, arithmetic and logic operations, fetch operation - word from memory, store operation - word in memory, example of complete instruction execution, example of branch instructions, Hardwired control unit design vs Micro-programmed control unit design, micro instructions, microprogramming.		
Case Study#7:	RISC and CISC architectures and design issues	
Case Study#8:	Study concepts of nano programming	
Unit 5	Memory System Hierarchy	6 Hours
Fundamentals of Memory Hierarchy and Memory parameters. Classifications of primary and secondary memories, Types of RAMS and ROM. Cache Memory: Cache organization: Direct-mapped, Set-associative, fully associative. Cache Write Policies: Write-through, Write-back. Cache Performance: Average Memory Access Time, Cost per bit calculation. Multi-level caches (L1, L2, L3). Main Memory and Interconnects: DRAM technologies and their characteristics. Secondary Memory- RAID- level 0 to level 6.		
Case Study#9:	Pentium4 cache organization	
Case Study#10:	Explore virtual memory and caching techniques in Android and iOS for efficient app performance.	
Unit 6	Input/output (I/O) Organization	5 Hours
Input/Output (I/O) Organization: Peripheral devices, I/O modules, I/O bus architectures, Data Transfer (Synchronous, Asynchronous), Modes of Data Transfer: Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA)-Intel 8237A-study.		
Case Study#11:	Study of Programmable Interrupt Controller 82C59A	
Case Study#12:	USB-C and Thunderbolt Interfaces Analyze modern high-speed I/O protocols and how they improve data transfer for peripherals.	
		Total Hours: 36
Textbooks:		
<ol style="list-style-type: none"> David A. Patterson and John L. Hennessy, "Computer Organization and Design RISC-V Edition: The Hardware/Software Interface," Morgan Kaufmann., 1st Edition, ISBN-0128122757, 978-0128122754 Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization," 5th Edition, McGraw-Hill. ISBN 0-07-112218-4 William Stallings, "Computer Organization and Architecture: Designing for Performance," 		

11th Edition, Pearson Education. ISBN: 978-93-560-6159.

Reference Books:

1. John P Hayes, "Computer Architecture and Organization" Tata McGraw Hill, 3rd Edition, ISBN-978-1259028564, 1259028569
2. Douglas Hall, Microprocessor and Interfacing Programming and Hardware, Tata McGraw Hill, 2nd Edition, ISBN- 0070257426, 978-0070257429
3. John L. Hennessy and David A. Patterson, "Computer Architecture: A Quantitative Approach", 6th Edition, Morgan Kaufmann. , ISBN: 0070273553.
4. M. Morris Mano, "Computer System Architecture", 3rd Edition, Pearson Education. ISBN: 0131755633.

E-Books/Online Learning Resources:

1. <https://os.ecci.ucr.ac.cr/ci0114/material/Stallings/Computer-Organization-Architecture-11th.pdf>
2. http://home.ustc.edu.cn/~louwenqi/reference_books_tools/Computer%20Organization%20and%20Architecture%2010th%20-%20William%20Stallings.pdf
3. [https://unidel.edu.ng/focelibrary/books/Computer%20Organisation%20and%20Architecture%20by%20Smruti%20Ranjan%20Sarangi%20\(z-lib.org\).pdf](https://unidel.edu.ng/focelibrary/books/Computer%20Organisation%20and%20Architecture%20by%20Smruti%20Ranjan%20Sarangi%20(z-lib.org).pdf)
4. https://mu.ac.in/wp-content/uploads/2021/03/COA_Full.pdf
5. <https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorial/>
6. <https://github.com/getvmio/free-computer-architecture-resources>

Online Courses:

MOOC/ Video lecture available:

1. Computer Architecture and Design
 - a. Instructor: Prof.Dr. P. SeshuBabu (NIT, Chennai)
 - b. Covers fundamental components, ability to describe and analyze the principle of computer organization.
 - c. Link: https://onlinecourses.swayam2.ac.in/ntr25_ed43/preview
2. Computer Architecture and Organization
 - a. Instructor: By Prof.IndraniI Sengupta, Prof.KamalikaDatta (IIT Kharagpur)
 - b. The concepts and challenges of Reduced Instruction Set Computer (RISC) platform.
 - c. Link: https://onlinecourses.nptel.ac.in/noc25_cs154/preview
3. GPU Architectures and Programming
 - a. Instructor: Prof.SoumyajitDey (IIT Kharagpur)
 - b. Covers optimization techniques relevant to both CUDA and OpenGL.
 - c. Link: https://onlinecourses.nptel.ac.in/noc25_cs37/preview

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	1	-	-	-	-	-	-	-	-	2
CO2	3	3	2	2	-	-	-	-	-	-	2
CO3	3	3	2	-	-	-	-	-	-	-	2
CO4	3	3	2	1	-	-	-	-	-	-	2
CO5	3	2	2	-	-	-	-	-	-	-	2
CO6	3	2	-	-	-	-	-	-	-	-	2

Program	B.Tech (Computer Science & Engineering)			Semester	III			
E/E/M	Entrepreneurship & Start-up Studio			Code	DP0402042 5306			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
2	2	--	--	50	--	--	--	50
Prerequisite:								
1) Familiarity with internet tools and digital platforms								
2) Communication and presentation skills								
Other Courses in this Series:								
1) NIL								
Course Objectives are not mentioned								
<ul style="list-style-type: none"> To explore various entrepreneurial career paths and provide foundational knowledge for launching new ventures. To equip students with tools and techniques for ideation, validation, and transforming innovative ideas into viable startups. To enable students to identify appropriate funding sources and effectively approach funding agencies for capital acquisition. To develop skills for preparing comprehensive business plans, operational models, and execution strategies. To empower students to identify real-world problems, evaluate market opportunities, and initiate their own entrepreneurial ventures. 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Identify and explore various entrepreneurial paths and acquaint them with the essential knowledge of starting new ventures.								
CO2: Apply appropriate tools and techniques for generating, testing and developing innovative start-up ideas into successful enterprises.								
CO3: Analyze different funding mechanisms and demonstrate the ability to approach suitable funding agencies for raising capital.								
CO4: Design and construct detailed business plans, including operational and financial strategies, for new entrepreneurial ventures.								
CO5: Evaluate business to start his/her own business by understanding the opportunities, targeting the larger problem to be addressed.								
Course Contents								
Unit 1	Introduction of Entrepreneurship							6 Hours
Concept and Need of Entrepreneurship Development, Concepts of Entrepreneur, Entrepreneur/Corporate Entrepreneur comparative study, Entrepreneurship Career opportunities, Entrepreneurship as a style of management, The Entrepreneur's Role, Task and Personality traits, Entrepreneurship in emerging economies.								
Case Study#1:	Study success story of Student Innovators or Entrepreneurs							
Case Study#2:	Watch some Entrepreneurial Films like - The Founder, The Social Network, Steve							

	Jobs, The Wolf of Wall Street and The Pursuit of Happiness.	
Unit 2	Idea Generation and Prototyping	6 Hours
	Techniques for brainstorming and generating ideas (Design Thinking) - Criteria for evaluating innovative ideas, Market research and feasibility analysis, Identifying potential customers and partners, How to conduct survey and design survey questionnaire , Developing and Testing Prototypes - The purpose and importance of prototyping, Different prototyping methods and tools, Creating low-fidelity and high-fidelity prototypes, User testing and feedback collection, Iterating and refining prototypes based on feedback.	
Case Study#3:	Apply design thinking methodologies involving brainstorming, prototyping, and user feedback for innovative product development.	
Case Study#4:	Prototype Designing Process and Tools for Development of Minimum Viable Product (MVP), Exposure visit to 3D Printing/vacuum casting/fast prototyping facility.	
Unit 3	Process and Planning	6 Hours
	Understanding Technology Readiness Level (TRL), Manufacturing Readiness Level (MRL) and Investment Readiness Level (IRL) Stages & Implications in Innovation Development, Innovation Management Process, Types of IPR, Patent Publication Process. Introduction to IPR- types, criteria, validity, process, IPR experts connect.	
Case Study#5:	Innovation Development vs. Innovation Management	
Case Study#6:	Design Patent draft of Innovation Development Idea	
Unit 4	Business Plans and Execution	6 Hours
	Business Plan Development: Components and Stages in Business Plan Development, Validation of Business Opportunity: Application of Market Research Tools at the Early Stage of Startup. Pitch Deck - Components, Pitch Deck Building, Pitch Format & Practice, one sentence pitch.	
Case Study#7:	Business model of Entrepreneur vs. Entrepreneur Business model Types	
Case Study#8:	Study Pitch Deck with real-life example	
Unit 5	Start-up and its Working	6 Hours
	Start-ups, Definition (about DPIIT), requirements and benefits, company act, opportunities and schemes, start-up registration process., Incubators and accelerators- roles, responsibilities and partnership, infrastructure. Funding, funding types , government schemes, Investments – need, sources, VC, angel investments. Start-up life cycle., Risk Management Strategies Predicting, Calculating and Mitigating Enterprise Risks, and Overcoming Financial Risks Project Monitoring and Controlling, Bouncing Back from Failures, Start-up Audit.	
Case Study#9:	Application of Market Research Tools at the Early Stage of Startup	
Case Study#10:	Study the various funding government and state government scheme	
	Total Hours: 30	
Text Books:		
<ol style="list-style-type: none"> 1. Rishiksha T. Krishnan, Vinay Dabholkar, “8 Steps to Innovation: Going from Jugaad to Excellence”, Collins India, 2013, ISBN-13: 978-9350293584 2. National Student and Faculty Startup Policy 2019. Government of India. Pavan Soni, “Design Your Thinking: The Mindsets, Toolsets and Skill Sets for Creative Problem Solving”, Penguin Random House India Pvt. Ltd., 2020, ISBN-13: 978-0670094097. 		

3. Prof. Rupinder Tewari, Ms. Mamta Bhardwaj, "Intellectual Property: A Primer for Academia", Publication Bureau, Panjab University, Chandigarh, 2021 (ISBN13: 8185322929)
<https://dst.gov.in/sites/default/files/E-BOOK%20IPR.pdf>

Reference Books:

1. P. Saravanel, "Entrepreneurial Development", Margham Publications, Chennai, 2020.
2. Prasanna Chandra, "Project Preparation, Appraisal, Implementation", Tata McGraw Hill, New Delhi, 2019.
3. N.P. Srinivasan, C.B. Gupta, "Entrepreneurial Development", Sultan Chand & Sons, 2017.

E-Books/Online Learning Resources:

1. P. Alloysius Augustine Ogbe, "Fundamentals of Entrepreneurship Development", available online, ISBN: N/A – covers foundational concepts, entrepreneurship traits, startup planning and institutional support.
2. https://www.researchgate.net/profile/Alloysius-Ogbe/publication/334863365_Fundamentals_of_Entrepreneurship_Development/
3. <https://www.startupindia.gov.in/>
4. https://onlinecourses.swayam2.ac.in/aic23_ge17/preview

Online Courses:

1. Innovation and Startup Policy
 - a. Instructor: Prof. Rahul K. Mishra (IILM Institute for Higher Education)
 - b. Covers Policy framework for innovation and startups, Linking innovation with IPR.
 - c. Link: https://onlinecourses.swayam2.ac.in/imb20_mg22/preview
2. Innovation, Business Models and Entrepreneurship
 - a. Instructor: Prof. Rajat Agrawal and Prof. Vinay Sharma (IIT Roorkee)
 - b. Covers Innovation and creative thinking, Technology incubation and forecasting.
 - c. Link: https://onlinecourses.nptel.ac.in/noc19_mg55/preview
3. Design Thinking – A Primer
 - a. Instructor: Prof. Ashwin Mahalingam & Prof. Bala Ramadurai (IIT Madras)
 - b. Covers Introduction to Design Thinking, Empathize phase & Customer Journey Mapping.
 - c. Link: https://onlinecourses.nptel.ac.in/noc22_mg32/preview
4. Introduction to Intellectual Property
 - a. Instructor: KAPILA IPR Committee (IIT Kharagpur)
 - b. Covers Copyrights, Patents, Trademarks.
 - c. Link: https://onlinecourses.swayam2.ac.in/aic21_ge20/preview

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1											
CO2											
CO3											
CO4											
CO5											
CO6											

Program	B.Tech (Computer Science & Engineering)			Semester	III			
VEC	Universal Human Values			Code	DP04020425307			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
2	2	--	--	50	--	--	--	50
Prerequisite:								
<ol style="list-style-type: none"> 1. Basic Communication Skills 2. Fundamental Understanding of Society and Ethics 3. Openness to Self-Reflection 4. General Life Skills Learned in First Year 5. Basic teamwork and interpersonal skills 								
Other Courses in this Series:								
NIL								
Course Objectives are not mentioned								
<ul style="list-style-type: none"> • Develop self-awareness and clarity of values. • Cultivate ethical decision-making and responsibility. • Understand relationships and human values. • Apply values in computing and teamwork. • Strengthen harmony with self, family, society, and nature. 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Identify personal values and strengths.								
CO2: Demonstrate empathy and communication.								
CO3: Apply ethical reasoning to technology.								
CO4: Analyze societal and environmental responsibilities.								
CO5: Engage in value-based real-life activities.								
Course Contents								
Unit 1								6 Hours
Case Study#1:								
Case Study#2:								
Unit 2								6 Hours
Case Study#3:								
Case Study#4:								
Unit 3								6 Hours
Case Study#5:								
Case Study#6:								
Unit 4								6 Hours

Case Study#7:											
Case Study#8:											
Unit 5										6 Hours	
Case Study#9:											
Case Study#10:											
											Total Hours: 30
Text Books:											
4.											
Reference Books:											
E-Books/Online Learning Resources:											
1.											
Online Courses:											
The CO-PO Mapping Matrix											
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1											
CO2											
CO3											
CO4											
CO5											
CO6											

Program	B.Tech. (Computer Science & Engineering)			Semester	III			
PCC	Data Structure and Algorithms Lab			Code	DP04020425308			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
1	--	--	2	--	--	--	25	25
Prerequisite: Problem Solving through Programming								
Other Courses in this Series: Data Structure and Algorithms								
Course Objectives: <ol style="list-style-type: none"> To develop a strong foundation in algorithm analysis To understand the role of abstract data types (ADTs) in software design and modular programming. To learn core data structures such as arrays, linked lists, stacks, queues, trees, and graphs. To implement and analyze searching, sorting, and hashing techniques for efficient data handling To identify and use the most suitable data structures and algorithms for practical implementation 								
Course Outcomes: On completion of the course, students will be able to: <p>CO1: Analyze various data structures and algorithmic strategies to evaluate their efficiency in terms of time and space complexity.</p> <p>CO2: Design optimized searching, sorting, and traversal techniques for structured data organization and retrieval.</p> <p>CO3: Apply appropriate linear and non-linear data structures such as stacks, queues, linked lists, trees, and graphs to solve real-world problems.</p> <p>CO4: Create efficient solutions for dynamic data management, including memory allocation, indexing, and data representations.</p> <p>CO5: Develop algorithms for hierarchical data management, dynamic memory handling, indexing.</p> <p>CO6: Implement solutions to real world problems using advanced data structures like Graphs and Optimal binary trees.</p>								
Suggested List of Laboratory Experiments/Assignments Assignments from all Groups (A, B, C, D, E, and F) are compulsory Programming Tools Recommended: C/C++/ Python								
Sr. No.	Group-A (Unit-1 Introduction to Data Structures and Algorithm Analysis)							
	Solve Any One:							
1)	<ol style="list-style-type: none"> Write a program for Polynomial Calculator to perform ADTs, addition, subtraction, and evaluation of polynomials entered by user. Write a program to compute the reverse of n number. Calculate time and time module. Also Find out time and space complexity manually. Write a program to find the address of an element in 2D array using Row-Major and Column-Major. 							

Sr. No.	Group-B (Unit-2 Searching and Sorting)
2)	<p>Solve Any Two:</p> <p>A. Write a program to store floating numbers in array in random order. Use Linear Search and Sentinel Search for searching an element and compare its complexity.</p> <p>B. Write a program to perform selection sort. Show its passes and compute its complexity.</p> <p>C. Write a Program to sort an array using Max Heap. Analyse the time complexity of Max Heap sort with Insertion sort.</p> <p>D. Write a Program to insert an element into hash table using division modulo method.</p>
Sr. No.	Group-C (Unit-3 Stack and Queue)
3)	<p>Solve Any Two:</p> <p>A. Write a program to compute transpose of sparse matrix using fast transpose.</p> <p>B. Write a program to convert the expression infix to postfix. Perform its evaluation using stack. Use suitable data.</p> <p>C. Write a program to perform insert and delete operations using circular queue. Use Job queue for adding job and deleting job from the queue.</p> <p>D. Write a program to implement Priority Queue. Consider an array to insert elements based on highest priority and delete elements with the lowest priority.</p>
Sr. No.	Group-D (Unit-4 Linked List)
4)	<p>Solve Any Two:</p> <p>A. Write a program to differentiate static memory and dynamic memory allocation. Use appropriate data structure.</p> <p>B. Write a program to search an element in Singly linked list. Display its location if element found.</p> <p>C. Write a program to perform functions to delete a node from:</p> <ul style="list-style-type: none"> • The beginning • The given position • The end of the doubly linked list <p>D. Write a program to define and display a generalized linked list. Use appropriate data.</p>
Sr. No.	Group-E (Unit-5 Tree)
5)	<p>Solve Any One:</p> <p>A. Write a program to convert a binary tree into threaded binary tree. Compare time and space complexity.</p> <p>B. Write a program to traverse a binary tree threaded using in-order threads, without using stack or recursion.</p> <p>C. Write a program to create a Binary Search Tree (BST) by inserting values one by one, and then perform following operations on it.</p> <ol style="list-style-type: none"> i. Insert a new node in BST ii. Delete a node iii. Search for a given value iv. Find longest path
Sr. No.	Group-F (Unit-6 Graph and Symbol Table)
6)	<p>Solve Any One:</p> <p>A. Perform DFS and BFS traversals using adjacency matrix and adjacency list representations for the area around the college as a graph using important landmarks as nodes.</p> <p>B. implement Dijkstra's algorithm to find the shortest path from a given source node to all other nodes in a weighted graph (no negative weights).</p> <p>C. Write a program to implement Topological Sort using DFS or Kahn's algorithm for a Directed Acyclic Graph (DAG).</p>
Sr. No.	Mini Projects: Design and Develop any one Mini-Project from Sr. No. 1 to 2
1)	Hospital Queue Management system using Linked list.
2)	College Campus Navigation system using suitable Data structure

Text Books:

1. **Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed**, *Fundamentals of Data Structures in C*, 2nd Edition, Universities Press, 2008. ISBN: **978-8173716058**
2. **Aaron M. Tenenbaum, Yedidyah Langsam, and Moshe J. Augenstein**, *Data Structures Using C and C++*, 2nd Edition, Pearson Education, 2009. ISBN: **978-0131997462**
3. **Mark Allen Weiss**, *Data Structures and Algorithm Analysis in C*, 2nd Edition, Pearson Education, 2006. ISBN: **978-0321441461**

Reference Books:

1. **Seymour Lipschutz**, *Data Structures*, Schaum's Outlines Series, McGraw Hill, 2014. ISBN: **978-0070601681**
2. **Debasis Samanta**, *Classic Data Structures*, 2nd Edition, PHI Learning, 2009. ISBN: **978-8120337315**
3. **Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman**, *Data Structures and Algorithms*, 1st Edition, Pearson Education, 1983. ISBN: **978-0201000238**
4. **Y. Kanetkar**, *Data Structures Through C++*, 4th Edition, BPB Publications, 2023. ISBN : **978-9355511898**

E-Books/Online Learning Resources:

1. **Robert Lafore**, *Data Structures and Algorithms in Java*, 2nd Edition, Sams Publishing, 2002. ISBN: **978-0672324536**.
<https://www.amazon.in/Data-Structures-Algorithms-Robert-Lafore/dp/0672324539>
2. **George Heineman, Gary Pollice, Stanley Selkow**, *Algorithms in a Nutshell*, 2nd Edition, O'Reilly Media, 2016. ISBN: **978-1491948927**
<https://www.oreilly.com/library/view/algorithms-in-a/9781491912973/>

Online Courses:**1. Problem Solving through Programming in C**

- a. Instructor: Prof. Anupam Basu (IIT Kharagpur)
- b. Covers C programming basics and problem-solving techniques.
- c. Link: https://onlinecourses.nptel.ac.in/noc22_cs101/preview

2. Crash Course on Data Structure and Algorithms in C++

- a. Instructor: Great Learning Academy Team
- b. Covers fundamental data structures (arrays, linked lists, trees) and algorithmic problem-solving using C++.
- c. Link: <https://www.mygreatlearning.com/academy/learn-for-free/courses/crash-course-on-data-structure-and-algorithms-in-c>

3. Advanced Data Structures and Algorithms in C++

- a. Instructor: *Holczer Balazs (Udemy)*
- b. Covers advanced concepts of trees, graph algorithms, and dynamic programming in C++.
- c. Link: <https://www.udemy.com/course/advanced-data-structures-and-algorithms-in-c/>

4. Free Data Structures and Algorithms in C++

- a. Instructor: *Scaler Topics Team (Scaler)*
 - b. Covers comprehensive C++ programming with a focus on data structures like stacks, queues, trees, and sorting algorithms.
 - c. Link: <https://www.scaler.com/topics/course/cpp-data-structures/>

Software Tools, Simulation Platforms & Hardware Kits:

1. GCC (GNU Compiler Collection) – Standard C compiler for Linux
2. Code::Block Open-source C/C++ IDE with built-in compiler support. <https://www.codeblocks.org/>
3. OnlineGDB (<https://www.onlinegdb.com/>) – Online compiler and debugger.

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	2	1	-	-	-	-	-	1
CO2	3	3	3	2	1	-	1	-	-	-	1
CO3	2	3	3	2	1	1	1	-	1	-	1
CO4	3	3	3	2	2	1	1	-	1	1	1
CO5	3	3	3	2	1	1	1	-	1	1	1
CO6	2	2	3	2	1	1	-	-	1	1	-

Program	B.Tech. (Computer Science & Engineering)			Semester	III			
CEP /FP	Mini Project			Code	DP04020425310			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
2	--	--	4	--	--	--	50	50

Prerequisite:

- 1) Problem Solving through Programming

Other Courses in this Series:

- 1) Coding Café-OOP

Course Objectives:

- 1) To introduce the principles of problem-solving using Object-Oriented Programming (OOP)
- 2) To develop the ability to design secure and maintainable class structures using encapsulation, access modifiers, properties with getters and setters.
- 3) To enable students to implement various types of inheritance (single, multiple, multilevel, hierarchical, and hybrid)
- 4) To equip students with skills for handling runtime errors using exception handling constructs like try, catch, throw.
- 5) To introduce the concept of generic programming using templates and to enable the use of STL components such as containers, iterators.

Course Outcomes:**On completion of the course, students will be able to:**

CO1: Understand the fundamentals of problem-solving using Object-Oriented Programming (OOP), and distinguish between procedural and object-oriented paradigms through the use of classes, objects, and binding techniques in C++.

CO2: Apply core object-oriented principles such as **encapsulation, access control**, and the use of **getter/setter methods** to design secure and maintainable class structures, while understanding mutable and immutable object behavior.

CO3: Demonstrate the implementation of various **inheritance models** (single, multiple, multilevel, hierarchical, hybrid), along with **function overloading, overriding**, and the use of abstract classes, interfaces, and special keywords like this.

CO4: Implement robust programs with proper **exception handling** (try, catch, throw) and **file handling** (read, write, append) using C++ streams, adhering to best practices for error detection and recovery.

CO5: Utilize **templates, STL containers, iterators**, and **algorithms** to create generic, reusable, and efficient C++ programs, enhancing modularity and performance.

Suggested List of Laboratory Experiments/Assignments
Assignments from all Groups are compulsory

Group-A

- | | | |
|----|----|--|
| a) | i) | Create a class Car with data members like brand, model, and price. Write member functions to input and display car details. Use public and private access specifiers |
|----|----|--|

		properly.
	ii)	Create a class Student that counts how many objects are created using a static variable. Add a constructor and destructor to show when objects are created and destroyed. Include inline functions for setting and displaying student data.
	iii)	Create two classes named Account and Loan. Each class should have one private data member: Account should store the balance, Loan should store the loan amount. Write a function called displayFinancials() that is not a member of either class, but can access and display the private data of both classes using the friend function concept. Also, create a class called Auditor which can access the private data of both Account and Loan using the friend class concept.
b)	i)	Create a class Employee with private data members name and salary. Use getter and setter methods to access and modify these values. Demonstrate encapsulation by controlling access to salary (e.g., prevent setting negative salary).
	ii)	Create a class Counter with a mutable data member accessCount that is modified inside a const function. Show how mutable allows change even when other members are constant
	iii)	Create a C++ program for a student database (Name, Roll No, Class, Division) using: Constructor, Default & Copy Constructor, Destructor, Static Function, Friend Class, this pointer, Inline Function, and dynamic memory (new/delete).
Mini Projects (Implement any one)		
1)		Design and implement a Student Management System to demonstrate fundamental concepts of Object-Oriented Programming (OOP). The system should allow for the creation, updating, and viewing of student records using classes and objects. The project must utilize key OOP features such as encapsulation, inheritance, and polymorphism, while clearly showcasing the difference between procedural and object-oriented approaches.
2)		Design and implement an Employee Information System in C++ that demonstrates key Object-Oriented Programming principles, especially encapsulation and controlled access to data. The system should use private data members, and provide public getter and setter methods to access or modify employee information. It should also differentiate between mutable and immutable objects through specific class designs.
Group-B		
a)	i)	Create classes Person (name, age), Employee (salary), and Manager (department) to demonstrate single, multilevel, multiple, hierarchical, and hybrid inheritance. Show constructor–destructor order and use appropriate access specifiers.
	ii)	Implement a class Complex which represents the Complex Number data type. Implement the following 1. Constructor (including a default constructor which creates the complex number 0+0i). 2. Overloaded operator+ to add two complex numbers. 3. Overloaded operator* to multiply two complex numbers. 4. Overloaded << and >> to print and read Complex Numbers.
	iii)	Create classes User, Developer, and Learner, where Developer and Learner inherit from User as virtual base classes. Create a class Trainee that inherits from both Developer and Learner. Override a function display () in each class and call it using a User* pointer.
b)	i)	Create a class MathOperation with a function to divide two numbers. Handle divide-by-zero using try, catch, and throw. Use nested try-catch blocks, rethrow, and demonstrate use of standard exception classes.
	ii)	Create a class Student with members like name, rollNo, and marks. Write and read objects to and from a binary file using file stream classes. Handle file errors, use file pointers to update a specific record, and detect end-of-file.
	iii)	Create a class Student to write and read student records using binary file handling. Use exception handling to catch invalid marks (e.g., <0 or >100) and handle file I/O errors gracefully.
Mini Projects (Implement any one)		

a)	Design and implement a Vehicle Management System in C++ that demonstrates various types of inheritance (single, multiple, multilevel, hierarchical, hybrid) along with method overloading, overriding, use of this pointer, abstract classes, and interfaces . The system will allow users to manage different categories of vehicles (e.g., Car, Bike, Truck) and showcase object-oriented design and code reusability.
b)	Develop a Student Report Card System in C++ that manages student academic records using file input/output (I/O) operations. The system should demonstrate robust exception handling using try, catch, and throw statements, and adhere to error handling best practices . The system must support functionalities like reading, writing, and appending student data in files while handling possible runtime errors such as file not found, invalid input, or division by zero.
Group-C	
a)	i) Create a function template findMax() to return the maximum of two values. Create a class template SimpleCalculator<T> to perform addition, subtraction, multiplication, and division for int and float types.
	ii) Write a C++ program to store and manage a list of Book records (Book ID, Title, Author, Price) using a vector. Implement sorting by price and searching by title using STL algorithms.
Mini Project	
1)	Design and implement a Generic Library Inventory System in C++ that utilizes function and class templates to manage books, journals, and magazines generically. The system should leverage non-type template parameters where applicable and make use of STL containers (like vector, map), iterators, and algorithms to perform efficient data handling and searching.
Textbooks & Reference Material	
<ol style="list-style-type: none"> 1. E. Balagurusamy, Object Oriented Programming with C++, 8th Edition, McGraw Hill Education, ISBN: 9789352607990 2. Robert Lafore, Object-Oriented Programming in C++, 4th Edition, Pearson Education, ISBN: 9788131703839 3. Bjarne Stroustrup, The C++ Programming Language, 4th Edition, Addison-Wesley, ISBN: 9780321563842 	
Reference Books:	
<ol style="list-style-type: none"> 1. Herbert Schildt, C++: The Complete Reference, 4th Edition, McGraw Hill Education, ISBN: 9780070532465 2. Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, C++ Primer, 5th Edition, Addison-Wesley, ISBN: 9780321714114 3. Nicolai M. Josuttis, The C++ Standard Library: A Tutorial and Reference, 2nd Edition, Addison-Wesley, ISBN: 9780321623218 	
E-Books/Online Learning Resources:	
<ol style="list-style-type: none"> 1. https://www.cplusplus.com/ 2. https://www.geeksforgeeks.org/c-plus-plus/ 3. https://www.tutorialspoint.com/cplusplus/ 4. https://web.p.ebscohost.com/ehost/search/basic?sid=a1470841-3443-4c3e-94aa-493852e4faca@redis&vid=0&tid=2003EB 	
Online Courses:	
<ol style="list-style-type: none"> 1. The Complete C++ Developer Course – Udemy <ol style="list-style-type: none"> a. Instructor: Frank J. Mitropoulos b. Topics Covered: OOP, STL, Templates, File Handling, Real-World Projects c. Link: https://www.udemy.com/course/beginning-c-plus-plus-programming/ 2. Object Oriented Programming in C++ – NPTEL (Swayam) <ol style="list-style-type: none"> a. Instructor: Prof. Partha Pratim Das, IIT Kharagpur 	

- b. Topics Covered: Classes and Objects, Inheritance, Polymorphism, Exception Handling, File I/O, Templates
c. Link: https://swayam.gov.in/nd1_noc20_cs55

Software Tools, Simulation Platforms & Hardware Kits:

1. Code::Blocks
Open-source IDE for C/C++ with GCC compiler support
Link: <http://www.codeblocks.org/>
2. Dev-C++
Lightweight IDE for C++ with integrated compiler (MinGW)
Link: <https://sourceforge.net/projects/orwelldevcpp/>
3. Eclipse IDE for C/C++ Developers
Cross-platform IDE supporting advanced project features
Link: <https://www.eclipse.org/downloads/packages/>
4. OnlineGDB
Online compiler and debugger for C++
Link: https://www.onlinegdb.com/online_c++_compiler

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	1	-	2	-	-	-	-	-	1
CO2	3	3	2	-	2	-	-	-	-	-	1
CO3	3	2	3	-	2	-	-	1	-	-	1
CO4	2	3	2	1	3	-	-	-	-	-	1
CO5	3	2	2	-	3	-	-	-	-	-	2

Program	B.Tech. (Computer Science & Engineering)			Semester	III			
MNC	Environmental Studies			Code	DP0420425311			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
00	2	--	--	--	--	--	--	--
Prerequisite: Science, Chemistry, Geography								
Other Courses in this Series: Nil								
Course Objectives: <ol style="list-style-type: none"> To understand the importance of use of natural resources such as water, soil, air, forests, and wildlife, and examine practices that promote conservation. To Gain a basic understanding of ecosystems and the interrelationship between living organisms and their environment. To develop the ability to critically analyse current environmental issues such as pollution, climate change, deforestation, and loss of biodiversity. To understand the impact of human activities such as industrialization, urbanization, and agricultural practices on the environment. To develop practical strategies for eco-friendly living through waste reduction, recycling, energy conservation, and awareness of environmental ethics and laws. 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Identify the problems related to natural resource depletion and sustainable practices.								
CO2: Analyze the dynamic nature of ecosystems and their importance in maintaining environmental balance.								
CO3: Understand the significance of biodiversity in maintaining ecological balance and supporting life on Earth.								
CO4: Evaluate pollution types and control measures and assess environmental laws and global protocols.								
CO5: Apply ethical, social, and green technology approaches for environmental sustainability.								
Course Contents								
Unit 1	Fundamentals of Environmental studies and Sustainability						6 Hours	
Definition, scope and importance of Environment Science, Need for public awareness, Natural resources and associated problems. Types of natural resources: renewable and non-renewable. Water resources, forest resources, mineral resources, food and land resources. Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyles, SDGs for environmental Science, Overexploitation and consequences. Conservation practices and management.								
Case Study#1:	The Amazon Rainforest has been facing large-scale deforestation due to							

	logging, agriculture, and cattle ranching. Analyse the environmental, social, and economic impacts of deforestation in the Amazon. Propose sustainable solutions that balance economic development and environmental conservation in the region	
Case Study#2:	Infosys Campus – Electronic City, Bangalore: Infosys implemented large-scale rainwater harvesting systems across its Bangalore campus, collecting and storing rainwater from rooftops and open areas.	
Unit 2	Ecosystem	6 Hours
Definition, scope and importance of environmental science. Components of the environment – atmosphere, hydrosphere, lithosphere, biosphere. Ecosystem – structure and functions. Types of ecosystems – forest, grassland, desert, aquatic. Climate-Resilient Ecosystems, Urban Ecosystems, AI/ML applications in ecosystem health prediction.		
Case Study#3:	NASA’s “Air Quality Monitoring” via AI and Satellites: NASA uses ML algorithms to analyze troposphere NO ₂ levels globally, influencing urban planning and health advisories.	
Case Study#4:	Smart Grazing Management in Indian Grasslands with GPS and ML: Tracking cattle movement and vegetation cover with satellite + IoT to prevent overgrazing.	
Unit 3	Biodiversity and its conservation	6 Hours
Introduction- genetic, species and ecosystem diversity, Bio geographical classification of India, Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values, Biodiversity at global, National and local levels, India as a mega diversity nation, Hot-spots of biodiversity, Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, Endangered and endemic species of India, Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.		
Case Study#5:	A national park in India is experiencing a rapid decline in genetic diversity among its tiger population due to habitat fragmentation. Analyse the consequences of reduced genetic diversity on the survival of the species. How can conservation strategies such as wildlife corridors or genetic management help protect this species from extinction?	
Case Study#6:	Ayurvedic Plants in Western Ghats – Role in Medicine: Database creation of medicinal plant species with APIs for search and categorization.	
Unit 4	Environmental Pollution and climate Change	6 Hours
Definition, Cause, effects and control measures of: -Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, nuclear hazards, Solid waste Management: Causes, effects and control measures of urban and industrial wastes, Role of an individual in prevention of pollution, Solid waste management – types and disposal techniques. E-waste and plastic waste. Climate change – global warming, greenhouse effect, acid rain, ozone layer depletion. Disaster management: types (natural and manmade), disaster preparedness and mitigation strategies.		
Case Study#7:	Smart City Pune: Environmental Sensing and Urban Heat Island Tracking: IoT sensors installed in key areas monitor temperature, air quality, and noise in real time.	

Case Study#8:	Swacch Bharat App and Smart Bin System: Smart bins in Indore send alerts when full; waste collection is digitally tracked.	
Unit 5	Environmental Social Issues, Legislation, Ethics and Green Technologies	6 Hours
Environmental protection acts – Water Act, Air Act, Environment Protection Act, Wildlife Protection Act, Forest Conservation Act. Role of Central and State Pollution Control Boards. Environmental ethics – issues and responsibilities. Corporate social responsibility (CSR) and environment. Concept of green technologies: clean energy, green buildings, waste to energy. Role of ICT and IoT in environmental monitoring. Environmental impact assessment (EIA).		
Case Study#9:	Tata Power’s Solar Micro grid Project in Rural India: CSR initiative promoting clean energy in remote villages.	
Case Study#10:	Digital Water Management-Jal Jeevan Mission: Use of IoT sensors to monitor tap water supply in rural India.	
Total Hours:30		
Field work/Visit		
<ul style="list-style-type: none"> • Visit to a local area to document environmental assets river/forest/grassland/hill/mountain to understand the ecological characteristics of the site. • Visit to a local polluted site-Urban/Rural/Industrial/Agricultural to identify the real time problems. • Visit to the Botanical Garden and study the common plants, insects, birds and simple ecosystems to recognize the importance of conservation of native and medicinal plant species. 		
Textbooks:		
<ol style="list-style-type: none"> 1. Erach Bharucha, “Environmental Studies”, UGC, University Press, ISBN: 9788173715402 2. Benny Joseph, “Environmental Studies”, Tata McGraw-Hill Education, ISBN: 9780070621176 3. Anubha Kaushik and C.P. Kaushik, “Environmental Studies”, New Age International Publishers, ISBN: 9788122435069. 		
Reference Books:		
<ol style="list-style-type: none"> 1. R. Rajagopalan, “Environmental Studies: From Crisis to Cure”, 4th Edition, Oxford University Press, ISBN: 9780198067643 2. Suresh K. Dhameja, “Environmental Engineering and Management”, S.K. Kataria & Sons, 2nd Edition, ISBN: 9789384007724 3. Gilbert M. Masters, “Introduction to Environmental Engineering and Science”, 3rd Edition, Pearson, ISBN: 978013148193 		
E-Books/Online Learning Resources:		
<ol style="list-style-type: none"> 1. Website:https://education.nationalgeographic.org 2. Website:https://www.unep.org 3. Website:https://enviroliteracy.org 		
Online Courses:		
<ol style="list-style-type: none"> 1) Introduction to Environmental Science and Ecosystems <ol style="list-style-type: none"> a) Instructor: Dr. Monica Jain of Devi Ahilya Vishwavidyalaya b) Covers Multidisciplinary nature, sustainability, ecosystem basics, ecological succession, ecosystem types. c) Link: Swayam - Environmental Studies 		

- 2) Sustainability, Climate Change and Resource Management
- Instructor: Muhammad Abdul Rahim (Finance & Energy Risk professional).
 - Covers Climate risk management, Energy markets overview
 - Link:** Udemy - Environmental Science and Sustainability
- 3) Environmental Policies, Governance and Legal Frameworks
- Instructor: Donald Hornstein (environmental law specialist)
 - Covers Pollution prevention, compliance, environmental policy & governance structures, legal proceedings.
 - Link:** <https://www.coursera.org/learn/environmental-law>

The CO-PO Mapping Matrix:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	2	-	-	-	3	2	-	-	-	2
CO2	2	3	-	2	-	3	2	-	-	-	2
CO3	2	2	-	-	-	3	2	-	-	-	2
CO4	2	3	1	2	-	3	3	-	-	1	2

1. Multi-disciplinary Minors (MDM)

Sr. No.	Title of the Multidisciplinary Minor (MDM)	Name of the offering Department/School/Program	Credits
1.	Minor in IoT and Embedded Systems	ENTC	14
2.	Minor in Hybrid Vehicles	MECH	14
3.	Minor in Mechanical Design & CAD	MECH	14
4.	Minor in Robotics and Automation	MECH	14

Minor in IoT and Embedded Systems								
Sr. No.	Semester	Course Code	Course Title	Engagement Hrs/Week (Teaching Scheme)				Credits
				L	T	P	Total Hrs	
1.	III	DP04020425304	Embedded System Fundamentals	3	0	2	5	4
2.	IV		IoT Architecture and Protocols	3	0	0	3	3
3.	V		Advanced Embedded Systems	3	0	2	5	4
4.	VI		Industrial IoT and Edge Computing	3	0	0	3	3
Total Academic Engagement and Credits				12	0	4	16	14

Minor in Hybrid Vehicles								
Sr. No.	Semester	Course Code	Course Title	Engagement Hrs/Week (Teaching Scheme)				Credits
				L	T	P	Total Hrs	
1.	III		Fundamentals of Hybrid and Electric Vehicles	3	0	2	5	4
2.	IV		Embedded Systems and Communication Protocols in Hybrid Vehicles	2	0	2	4	3
3.	V		Vehicle Control and Software Systems	3	0	2	5	4
4.	VI		Data Analytics and AI for Intelligent Vehicles	1	0	4	5	3
Total Academic Engagement and Credits				9	0	10	19	14

Minor in Mechanical Design & CAD								
Sr. No.	Semester	Course Code	Course Title	Engagement Hrs/Week (Teaching Scheme)				Credits
				L	T	P	Total Hrs	
1.	III		Course-1: Fundamentals of Computer-Aided Design	3	0	2	5	4
2.	IV		Course-2: Generative Design and Topology Optimization	2	0	2	4	3
3.	V		Course-3: CAD Automation with AI and Python	3	0	2	5	4
4.	VI		Coure-4: Mini Project	1	0	4	5	3
Total Academic Engagement and Credits				9	0	10	19	14

Minor in Robotics and Automation								
Sr. No.	Semester	Course Code	Course Title	Engagement Hrs/Week (Teaching Scheme)				Credits
				L	T	P	Total Hrs	
1.	III		Principles of Robotics	3	0	2	5	4
2.	IV		Kinematics of Machinery	3	0	0	3	3
3.	V		Robot Programming & Simulation	3	0	2	5	4
4.	VI		Industrial Automation and Control System	2	1	0	3	3
Total Academic Engagement and Credits				11	1	4	16	14

Program Elective Courses (PEC)

- Students are urged to register for the program elective courses offered under Semester VII and VIII through SWAYAM or NPTEL Portal for a minimum of 12 weeks & 3 credits, subject to prior approval from the department. Previously completed courses under PCC, PEC, Honors, Minors or any other category should not be selected.

Baskets of Program Elective Courses

Elective No.	Semester	Course Title	Remark	Credits
III	VII	<ul style="list-style-type: none"> Generative AI Introduction to Large Language Models (LLMs) 	Choose any one through SWAYAM/ NPTEL of minimum 12 weeks and 3 credits course.	3
IV		<ul style="list-style-type: none"> Quantum Computing Reinforcement Learning Privacy and Security in Online Social Media Augmented and Virtual Reality Edge Computing 	Choose any one through SWAYAM/ NPTEL of minimum 12 weeks and 3 credits course.	3
V	VIII	<ul style="list-style-type: none"> High-Performance Computing IoT (Internet of Things) Blockchain Technology Cybersecurity and Forensics 	Choose any one through SWAYAM/ NPTEL of minimum 12 weeks and 3 credits course.	3
VI		<ul style="list-style-type: none"> Deep Learning Big Data Analytics Data Science and Analytics Data Modeling and Visualization Pattern Recognition and Optimization 	Choose any one through SWAYAM/ NPTEL of minimum 12 weeks and 3 credits course.	3

Elective No.	Semester	Course Title	Remark	Credits
		<ul style="list-style-type: none"> ▪ Natural Language Processing ▪ Cloud Computing ▪ Distributed Systems ▪ Agile Software Development ▪ Data Mining and Warehousing ▪ Business Analytics ▪ Advanced Computer Networks ▪ Software Testing and Quality Assurance ▪ Information Retrieval ▪ Games and Information ▪ Advanced Computer Graphics ▪ Advanced Computer Architecture ▪ GPU Architectures and Programming ▪ Parallel Computer Architecture ▪ Introduction to Industry 4.0 and Industrial IoT ▪ Object-Oriented Modeling and Design ▪ Cryptography and Information Security ▪ Web Technologies ▪ Human-Computer Interface ▪ Digital Signal and Image Processing ▪ Advanced Data Structures ▪ Embedded Systems ▪ Ethical Hacking ▪ Quantum Algorithms and Cryptography ▪ Digital Logic and Microprocessor ▪ Social Networks 		

Optional Honors in Computer Science & Engineering

Sr. No.	Title of the Honor in CSE	Name of the offering Department/School/Program	Credits
1.	Honors in <i>Cloud Computing</i>	CSE	20
2.	Honors in <i>Full-Stack Development</i>	CSE	20
3.	Honors in <i>Internet of Things</i>	CSE	20
4.	Honors in <i>Artificial Intelligence and Machine Learning</i>	CSE	20
5.	Honors in <i>Information Security/Cyber Security</i>	CSE	20
6.	Honors in <i>Blockchain Technology</i>	CSE	20
7.	Honors with Research	CSE	20

Semester-wise Details of the Courses of B.Tech. (CSE)

SEMESTER 4

Catalog Course Code	Course Code	Course Type	Course Title	EngagementHrs/week (Teaching Scheme)				Credits	Continuous Assessment		Term End Examination		Total
				L	T	P	Total		Theory	Practical	Theory	Practical	
STENCS071V0	DP04020425401	PCC	Computer Networks and Security	3	-	-	3	3	45	0	30	0	75
STENCS026V0	DP04020425402	PCC	Theory of Computation	3	-	-	3	3	45	0	30	0	75
STENCS027V0	DP04020425403	PCC	Advanced Programming in WebX	1	-	-	1	1	25	0	0	0	25
STENCS194V0	DP04020425404	MDM	IoT Architecture and Protocols	3	-	-	3	3	45	0	30	0	75
STENCS195V0	DP04020425405	OE	Applied Machine Intelligence for Fintech	3	-	-	3	3	45	0	30	0	75
STENCS028V0	DP04020425406	VSEC	WebX Design and Development Lab	-	-	4	4	2	0	30	0	20	50
STENCS029V0	DP04020425407	PCC	Computer Networks and Security Lab	-	-	2	2	1	0	0	0	25	25
STENCS030V0	DP04020425408	AEC	Effective Technical Communication	2	-	-	2	2	50	0	0	0	50
Entrepreneurship/Economics/Management Courses (E/E/M) (Choose Any One)													
STENCS196V0	DP04020425409	E/E/M	Machine Intelligence for Fintech System Lab	-	-	4	4	2	0	50	0	0	50
VEC and MNC Courses													
STENCS197V0	DP04020425410	VEC	Constitution of India	2	-	-	2	2	30	0	20	0	50
HLHMCS006V0	DP04020425411	MNC	Emotional Well Being	2	-	-	2	-	0	0	0	0	-

Total Academic Engagement and Credits	19	-	10	29	22	285	80	140	45	550
--	-----------	----------	-----------	-----------	-----------	------------	-----------	------------	-----------	------------

Note: 1 credit= 25 marks

L: Lecture, **T:** Tutorial, **P:** Practical



Program	B. Tech (Computer Science & Engineering)					Semester	IV		
PCC	Computer Networks and Security					Code	DP04020425401		
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks					
Credits	Lectures	Tutorials	Practical	CA		TEA		Total	
				Theory	Practical	Theory	Practical		
3	3	--	--	45	--	30	--	75	
Prerequisite: Fundamentals of Computer Engineering									
Other Courses in this Series: Advanced Computer Networks and Security									
Course Objectives: <ol style="list-style-type: none"> 1: Understand the fundamentals of computer networks, layered architectures, protocols, and standards. 2: Analyze data communication mechanisms across Physical, Data Link, Network, and Transport layers. 3: Design and evaluate routing, congestion control, and transport layer mechanisms for efficient data transfer. 4: Understand application layer protocols and network services used in real-world systems. 5: Gain foundational knowledge of network security principles, threats, cryptographic concepts, and security mechanisms. 									
Course Outcomes: On completion of the course, students will be able to: CO1: Explain the goals, architecture, reference models (OSI, TCP/IP), and physical layer concepts of computer networks. CO2: Analyze framing, error control, flow control, and MAC protocols in the data link layer. CO3: Apply network layer concepts including IP addressing, routing algorithms, congestion control, and IPv6. CO4: Analyze transport layer protocols (TCP/UDP), connection management, congestion control, and QoS mechanisms. CO5: Explain application layer protocols such as DNS, HTTP, FTP, email, and network management services. CO6: Understand security goals, threats, cryptographic basics, and security mechanisms in network, transport, and application layers.									
Course Contents									
Unit 1	Introduction to Computer Networks							08 Hours	

Introductory Concepts: Goals and applications of networks, Categories of networks, Organization of the Internet, ISP, Network structure and architecture (layering principles, services, protocols and standards), The OSI reference model, TCP/IP protocol suite, Network devices and components. Physical Layer: Network topology design, Types of connections, Transmission media, Signal transmission and encoding, Network performance and transmission impairments, Switching techniques and multiplexing.		
Case Study#1:	Design and Analysis of a Small Office / Campus Network	
Case Study#2:	Impact of Physical Layer Design on Network Performance	
Unit 2	Data Link Layer	07 Hours
Link layer: Framing, Error Detection and Correction, Flow control (Elementary Data Link Protocols, Sliding Window protocols). Medium Access Control and Local Area Networks: Channel allocation, Multiple access protocols, LAN standards, Link layer switches & bridges (learning bridge and spanning tree algorithms).		
Case Study#3:	Selection of Data Link Devices in Enterprise Networks	
Case Study#4:	Implementation and Benefits of Virtual LANs (VLANs)	
Unit 3	Network Layer	08 Hours
Network Layer: Point-to-point networks, Logical addressing, Internet Protocols, Basic internetworking (IP, CIDR, ARP, RARP, DHCP, ICMP), Routing, forwarding and delivery, Static and dynamic routing, Routing algorithms and protocols, Congestion control algorithms, IPv6.		
Case Study#5:	Addressing and Internetworking in Enterprise Networks	
Case Study#6:	Routing and Congestion Control Analysis Using Simulation Tools	
Unit 4	Transport Layer	07 Hours
Transport Layer: Process-to-process delivery, Transport layer protocols (UDP and TCP), Multiplexing, Connection management, Flow control and retransmission, Window management, TCP Congestion control, Quality of service.		
Case Study#7:	Comparative Study of Transport Protocols for Web and Streaming Applications	
Case Study#8:	IC Protocol and Modern Transport Layer Evolution	
Unit 5	Application Layer	07 Hours
Application Layer: Domain Name System, World Wide Web and Hyper Text Transfer Protocol, Electronic mail, File Transfer Protocol, Remote login, Network management, Data compression, Cryptography – basic concepts.		
Case Study#9:	Analysis of Application Layer Protocols Using Network Protocol Analyzer	
Case Study#10:	DHCP and Application Layer Services in Large-Scale Networks	
Unit 6	Security in Computer Networks	08 Hours
Introduction, Need of Security, Security services, Security goals, Key Principles of Security, Threats and Vulnerabilities, Types of Attacks, ITU-T X.800 Security Architecture for OSI, Security Policy and mechanisms, Operational Model of Network Security, Security in Network, Transport and Application: Introduction of IPSec, SSL, HTTPS, S/MIME, Overview of IDS and Firewalls.		
Case Study#11:	Security Protocols Across OSI Layers	

Case Study#12:	a. Real-World Cyber Attacks and Defense Mechanisms b. Cryptography and Zero Trust Security in Modern Networks										
Total Hours: 45											
Text Books:											
1. Behrouz A. Forouzan, "Data Communications and Networking", 5th Edition, McGraw-Hill Education, ISBN: 978-0-07-337622-6 2. Andrew Tanenbaum, "Computer Networks", 5th Edition, Pearson, ISBN-13: 978-0132126953 ISBN-10: 0132126958. 3. Forouzan. TCP/IP Protocol Suite. 4th Edition. 4th Edition - 27 March 2009. ISBN-13: 978-0073376042 ISBN-10: 0073376043.											
Reference Books:											
1. James Kurose, Keith Ross, "Computer Networking: A Top-Down Approach", 7 th Edition, Pearson Education, ISBN: 978-0133594140 2. William Stallings, "Data and Computer Communications", 4th Edition, McGraw Hill, ISBN: 978-0070411838											
E-Books/Online Learning Resources:											
1. Computer Networking: Principles, Protocols and Practice, fourth edition https://4ed.computer-networking.info/syllabus/default/index.html 2. CCNA: Introduction to Networks, https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US 3. Computer Networks: A Systems Approach https://open.umn.edu/opentextbooks/Textbooks/771 4. Computer Network Tutorial https://www.geeksforgeeks.org/computer-networks/computer-network-tutorials/											
Online Courses:											
1. Computer Networks and Internet Protocol <ol style="list-style-type: none"> Instructor: Prof. Ajit Pal (IIT Kharagpur) Covers: Fundamentals of computer networks, TCP/IP protocol stack, routing, congestion control, socket programming, SDN, and network security. Link: https://nptel.ac.in/courses/106105080 2. Computer Networks <ol style="list-style-type: none"> Instructor: Prof. Sandip Chakraborty & Prof. Soumya Kanti Ghosh (IIT Kharagpur) Covers: Fundamentals of computer networks, TCP/IP protocol stack, routing, congestion control, socket programming, SDN, and network security. Link: https://onlinecourses.nptel.ac.in/noc25_cs15/preview 											
The CO-PO Mapping Matrix											
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	–	–	–	–	–	–	–	–	–
CO2	3	3	2	–	–	–	–	–	–	–	–
CO3	3	3	3	2	–	–	–	–	–	–	–
CO4	3	3	3	2	–	–	–	–	–	–	–
CO5	2	2	2	–	2	–	–	–	–	2	–
CO6	2	3	2	2	2	2	–	2	–	–	–



Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
PCC	Theory of Computation			Code	DP04020425402			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
3	3	--	--	45	--	30	--	75
Prerequisite: Discrete Mathematics								
Other Courses in this Series: Nil								
Course Objectives: <ul style="list-style-type: none"> • To introduce the foundational models of computation, such as finite automata, pushdown automata, and Turing machines. • To develop an understanding of formal languages, grammars, and their classification in the Chomsky hierarchy. • Explore PDA, Turing machines and computability principles for problem-solving efficiency. • Investigate computational complexity and its impact on algorithm design. 								
Course Outcomes: On completion of the course, students will be able to: <p>CO1: Understand formal languages, translation logic, and representation techniques; apply these concepts to design Finite Automata and its variants.</p> <p>CO2: Construct regular expressions for regular languages and analyze the properties of regular languages using the Pumping Lemma.</p> <p>CO3: Design and simplify context-free grammar (CFGs) to generate context free languages.</p> <p>CO4: Construct Pushdown Automata (PDA) to recognize context-free languages.</p> <p>CO5: Design Turing Machines to solve specific computational problems as per theoretical models.</p> <p>CO6: Analyze different classes of computational problems and evaluate the concepts of P, NP, NP-completeness, and reducibility.</p>								
Course Contents								
Unit 1	Introduction to TOC and Finite Automata						6 hours	

Introduction of Compiler, assembler, linker, loader. Finite Automata (FA): An informal picture of FA, Finite State Machine (FSM), Language accepted by FA, Definition of Regular Language. FA without output: Deterministic and Nondeterministic FA (DFA and NFA), epsilon- NFA and inter-conversion. Minimization of DFAs. FA with output: Moore and Mealy machines -Definition, models, inter-conversion.		
CaseStudy#1:Preposition Logic in AI	Objective: To help students understand the roles of compilers, assemblers, linkers, loaders, and the application of Finite Automata (FA) in system design, using a real-world example. Scenario: You are part of a software development team tasked with designing a basic vending machine that accepts coins (Rs. 1 and Rs. 2) and dispenses a drink when a total of Rs. 5 is collected.	
Unit 2	Finite Automata and Regular Languages	6 Hours
Introduction, Operators of RE, Precedence of operators, Algebraic laws for RE, Language to Regular Expressions, Equivalence of two REs. Conversions: RE to NFA, DFA, DFA to RE using Arden's theorem, Pumping Lemma for Regular languages, Closure and Decision properties of Regular languages. Myhill-Nerode theorem.		
Case Study#2: Regular Expression for Full Stack Development	Objective: To illustrate how Regular Expressions, their algebraic properties, and related finite automata concepts can be used to validate a basic password policy. Scenario: Your software firm is creating a login system. You're asked to implement a password validation module that accepts passwords following these rules: Must start with a capital letter (A–Z) Followed by 3 to 5 lowercase letters (a–z) Ends with exactly 2 digits (0–9) Examples of valid passwords: Aabc12,	
Simulation	https://www.asethome.org/fa/	
Unit 3	Context Free Grammar (CFG) and Context Free Language (CFL)	6 Hours
Context-Free Grammars (CFG) & Derivation Trees, Simplification of CFG: Eliminating ϵ -productions, unit productions, useless production, useless symbols. Normal Forms: Chomsky Normal Form, Greibach Normal Form, Pumping Lemma for CFG, Closure properties of CFL, Decision properties of CFL, Chomsky Hierarchy, Cock-Younger-Kasami Algorithm.		

Case Study#3:CFG for NLP	<p>Objective: To model and simplify a context-free grammar (CFG) for a drawing app where users input basic commands to control a virtual pen—like "draw line", "draw circle", "move up", etc.—and explore all associated theoretical aspects.</p> <p>Scenario:</p> <p>You are creating a parser for a drawing tool that accepts the following command patterns:</p> <p>"draw line"</p> <p>"draw circle"</p> <p>"move up"</p> <p>"move down"</p> <p>"move left"</p> <p>"move right"</p>	
Simulation	https://vyaakaran.vercel.app/	
Unit 4	Pushdown Automata	6 Hours
Introduction, Formal definition of PDA, Equivalence of Acceptance by Final State and Empty stack, Nondeterministic PDA (NPDA), PDA and Context Free Language, Equivalence of PDA and CFG		
Case Study#4:Structuring chatbot conversations	<p>Objective: To apply the concepts of PDA, acceptance mechanisms, and its relationship with context-free languages (CFLs) in designing a feature for a basic text editor.</p> <p>Scenario:</p> <p>You're building a simple text editor that provides syntax highlighting. One feature it must support is checking whether a given string of parentheses is balanced. The parentheses can be of one type: (and). The checker must accept strings like (), (()), (()()), and reject ((), ())(), etc.</p>	
Simulation	https://www.asethome.org/pda/	
Unit 5	Turing Machines and Computability	6 Hours
Introduction to Turing Machines (TM) ,Variants of Turing Machines (Multi-Tape, Non-Deterministic) Decidability & Undecidability ,The Church-Turing Thesis ,Recursive & Recursively Enumerable Languages ,Halting Problem & Unsolvability		
Case Study#5:Identifying infinite loops in software development	<p>Objective: To explore the limits of computability by modeling how a basic spell checker can—or cannot—be built using Turing Machines and to examine whether such problems are decidable or undecidable.</p> <p>Scenario:</p> <p>Imagine you're building a retro-style spell checker for a text editor that only accepts words from a specific dictionary of correct words (like {"cat", "dog", "dot"}). Your task is to validate whether a given string belongs to this dictionary.</p>	
Simulation	https://www.asethome.org/mathfoundations/asynchronous/	
Unit 6	Computational Complexity & Advanced Topics	6 Hours

Complexity Classes (P, NP, NP-Complete, NP-Hard) ,Reducibility & Polynomial Time Reductions Introduction to Space Complexity ,Hierarchy Theorems & Complexity Bounds ,Introduction to Approximation Algorithms ,Real-World Applications & Emerging Trends	
Case Studies #6: logistics and route optimization	Traveling Salesman Problem (TSP): Used in logistics and route optimization, where companies like FedEx and UPS optimize delivery routes to minimize costs. The problem is NP-hard, meaning no known polynomial-time solution exists. Explore Here: https://www.geeksforgeeks.org/types-of-complexity-classes-p-np-comp-np-hard-and-np-complete/
Total Hours:36	
Text Books:	
<ol style="list-style-type: none"> 1. Michael Sipser, Introduction to the Theory of Computation, 3rd Edition, Cengage Learning, ISBN: 978-1133187790 2. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Automata Theory, Languages, and Computation, 3rd Edition, Pearson, ISBN: 978-0321455369 3. Harry R. Lewis, Christos H. Papadimitriou, Elements of the Theory of Computation, 2nd Edition, Prentice Hall, ISBN: 978-0132624787 	
Reference Books:	
<ol style="list-style-type: none"> 1. Peter Linz, Introduction to Automata Theory & Formal Languages, 5th Edition, Jones & Bartlett Learning, ISBN: 978-9380108433 2. K. L. P. Mishra and N. Chandrasekaran, <i>Theory of Computer Science: Automata, Languages and Computation</i>, 3rd Edition, PHI Learning, ISBN: 978-8120332007 3. Dexter C. Kozen, Automata and Computability, 1st Edition, Springer, ISBN: 978-0387974657 4. Sanjeev Arora and Boaz Barak, "Computational Complexity: A Modern Approach", Cambridge University Press, ISBN: 0521424267 97805214242643. 5. Vivek Kulkarni, Theory of Computation, 1st Edition, Oxford University Press, ISBN: 978-0199464160 	
E-Books/Online Learning Resources:	
<ol style="list-style-type: none"> 1. https://link.springer.com/book/10.1007/978-981-97-6234-7 2. https://ocw.mit.edu/courses/18-404j-theory-of-computation-fall-2020/ 3. https://www.amazon.in/Theory-Computation-Concepts-Models-Applications-ebook/dp/B0DV5S24QH 	
Online Courses:	
<ol style="list-style-type: none"> 1. Computer Science: Algorithms, Theory, and Machines <ol style="list-style-type: none"> a. Instructor: Robert Sedgewick b. Focuses on computability, universality, and intractability, turing Machine c. Link: https://www.coursera.org/learn/cs-algorithms-theory-machines 	

2. Theory of Computation

a. Instructor: Prof. Michael Sipser

b. Topics include regular and context-free languages, decidable and undecidable problems, reducibility, recursive function theory, time and space measures on computation, completeness.

c. Link: <https://ocw.mit.edu/courses/18-404j-theory-of-computation-fall-2020/>

3. Theory of Computation

a. Instructor: Prof. Ragunath Tewari

b. Topic include Finite automata, Regular expression, CFG, PDA, TM etc.

c. Link: https://onlinecourses.nptel.ac.in/noc19_cs79/preview**The CO-PO Mapping Matrix**

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	3	2	-	-	-	2	2	2
CO2	3	3	3	2	3	-	-	-	2	2	2
CO3	3	3	2	3	3	-	-	-	2	2	2
CO4	3	3	2	3	2	-	-	-	2	2	2
CO5	3	2	3	2	3	2	2	3	2	3	3
CO6	3	3	3	2	2	2	2	2	2	2	2



Dnyaan Prasad Global University
Dr. D. Y. Patil Unitech Society
School of Technology and Research

Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
PCC	Advanced Programming in WebX			Code	DP04020425403			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
1	1	--	--	25	--	--	--	25
Prerequisite:								
1) Problem Solving through Programming, Database Management Systems.								
Other Courses in this Series:								
1) WebX Design and Development Lab								
Course Objectives:								
1. To provide fundamental knowledge of web technologies.								
2. To develop dynamic web applications using modern frameworks.								
3. To introduce backend integration and cloud deployment.								
4. To explore cutting-edge advancements in web technology.								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Apply fundamentals of web technologies to develop structured and responsive web pages.								
CO2: Develop interactive and dynamic web applications using modern frameworks.								
CO3: Apply and analyse backend integration for secure and efficient web applications.								
CO4: Apply and evaluate deployment techniques for optimized web application performance.								
Course Contents								
Unit 1	Fundamentals of Web Technology							4 Hours
Basics of Web Architecture – Covers the Client-Server Model, HTTP request-response cycle, methods (GET, POST), and the importance of HTTPS with SSL (Secure Sockets Layer) and TLS (Transport Layer Security) for security.								
Hypertext Transfer Protocol Secure (HTML5) – Introduces basic HTML tags, HTML Table, HTML List, HTML forms, and multimedia (<audio>, <video>).								
Cascading Style Sheets (CSS) – Basic CSS properties, Types of CSS, types of selectors, Covers Flexbox, Grid, and Responsive Design.								

Case Study#1:	Consider any online service platform in an organization and identify the different web components used, classify the HTML elements, list the CSS techniques applied for layout and responsiveness, and analyse the impact of web design choices on performance and user experience.	
Unit 2	Interactive Web Development	4 Hours
JavaScript Basics – Variables, Data Types, Functions, Control Statements. DOM & Event Handling & Form Validation – Event Listeners, Form Validation. Frontend Frameworks – Intro to React JS, Properties, State, Lifecycle Methods, API Calls.		
Case Study#2:	Create a student registration system for an educational institution and identify the various form fields required, apply appropriate HTML input types, use CSS for layout and styling, explain JavaScript-based validation to ensure input accuracy, and analyse the importance of client-side validation in enhancing user experience and reducing server load.	
Unit 3	Backend and API Integration	4 Hours
Servlet and JSP –Overview of servlets, hello world program in servlet, Overview of JSP Database Connectivity with servlet/ JSP –Connecting to MySQL/MongoDB. Database Connectivity with React - Connecting to Firebase/MySQL		
Case Study#3:	Study a simple course feedback system for an academic institution, identify the feedback parameters to be collected from students, examine the use of JSP and Servlet technologies for handling form submissions and backend processing, explore the role of a relational database in storing and retrieving feedback data, and analyse the significance of server-side scripting in dynamic web application development.	
Unit 4	Deployment & Future Web Technologies	4 Hours
Web Hosting & Deployment –Deployment of Web App, via Docker, GitHub, AWS basics, domain setup. Security Best Practices – Secure API Communication, authentication. Future of Web technology -Overview of DevOps, Overview ofIntegrate ML models in frontend/backend.		
Case Study#4:	Study an e-commerce platform architecture using cloud services like Firebase or AWS, identify the key functional modules such as user authentication, product catalogue, and order management, examine the use of real-time databases for interactive features, explore API-based content delivery, and analyse how cloud-based technologies support scalability and performance in modern web applications.	
Total Hours: 16 Hours		
Textbooks:		
<ol style="list-style-type: none"> 1. Jeffrey Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035 2. Jon Duckett, "HTML & CSS: Design and Build Websites", 1st Edition, Wiley, ISBN: 978-1118008188. 3. Marijn Haverbeke, "Eloquent JavaScript", 3rd Edition, No Starch Press, ISBN: 978-1593279509 		
Reference Books:		

1. Author TBD, "Modern Web Development with WebX", 1st Edition, Publisher TBD, ISBN: TBD
2. Mario Casciaro, "Node.js Design Patterns", 3rd Edition, Packt Publishing, ISBN: 978-1839214110
3. Mark Masse, "REST API Design Rulebook", 1st Edition, O'Reilly Media, ISBN: 978-1449310509
4. Adam Freeman, "Pro Angular", 5th Edition, Apress, ISBN: 978-1484264489

E-Books/Online Learning Resources:

1. <https://www.w3.org/html/>
2. [HTML, The Complete Reference http://www.htmlref.com/](http://www.htmlref.com/)
3. <http://www.tutorialspoint.com/>
4. <https://www.geeksforgeeks.org/web-technology/>
5. HTML & CSS by Jon Duckett (Sample Pages)," Visual approach to learning web designbasics",<https://bcs.wiley.com/he-bcs/Books?action=mininav&bcsId=6056&itemId=1118008189&assetId=283213>.
6. Fullstack React , " Hands-on guide to building apps using React + Redux + APIs", <https://www.fullstackreact.com/>

Online Courses:

1. **Web Technology**
 - a. Instructor: Dr. Ashutosh Kumar Bhatt
 - b. Covers Web Technology Fundamentals, HTML, CSS, JavaScript, JSP.
 - c. Link: https://onlinecourses.swayam2.ac.in/nou24_cs09/preview
2. **Introduction to Modern Application Development, IIT Madras**
 - a. Instructor: By Prof. Gaurav Raina, Tanmai Gopal
 - b. Covers development of single-page web applications, dependency injection, typescript, components, and directives with this Angular Course.
 - c. Link: <https://nptel.ac.in/courses/106106156>

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	–	3	1	–	–	2	–	1
CO2	3	2	3	–	3	1	–	1	2	–	1
CO3	3	3	2	2	3	1	2	–	2	1	2
CO4	2	2	2	2	3	1	–	1	2	2	2



Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
MDM	IOT Architecture and Protocols			Code	DP04020425404			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
3	3	--	--	45	--	30	--	75
Prerequisite:								
<ol style="list-style-type: none"> 1) Computer Organization and Architecture 2) Embedded Systems Fundamentals 								
Other Courses in this Series: NIL								
Course Objectives:								
<ol style="list-style-type: none"> 1. To understand the fundamentals of the Internet of Things (IoT) 2. To learn advances in IoT systems 3. To learn methodologies for IoT application development 4. To learn the IoT protocols, cloud platforms, and security issues in IoT 5. To understand real-world application scenarios of IoT along with its societal and economic impact using case studies and real-life examples 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Understand the fundamentals of the Internet of Things and apply IoT enabling technologies for developing IoT systems								
CO2: Apply design methodology for designing and implementing IoT applications								
CO3: Understand IOT data processing topologies								
CO4: Analyze IoT protocols for making IoT devices communicate								
CO5: Design cloud-based IoT systems								
CO6: Design and develop secure IoT applications								
Course Contents								
Unit 1	IOT Overview							7 Hours

<p>Introduction to Internet of Things (IoT): Definition, Characteristics of IoT, Vision, Trends in Adoption of IoT, IoT Devices, IoT Devices Vs Computers, Societal Benefits of IoT, Technical Building Blocks.</p> <p>Physical Design of IoT: Things in IoT, Interoperability of IoT Devices, Sensors and Actuators, Need for Analog / Digital Conversion.</p> <p>Logical Design of IoT: IoT functional blocks, IoT enabling technologies, IoT levels and deployment templates, Applications in IoT.</p>		
Case Study#1:	Smart Home Automation System	
Case Study#2:	Precision Agriculture for Sustainable Farming	
Unit 2	IoT: Design Methodology	8 Hours
<p>IoT Design Methodology: Steps, Basics of IoT Networking, Networking Components, Internet Structure, Connectivity Technologies, IoT Communication Models, IoT Communication APIs, and Sensor Networks. Four pillars of IoT: M2M, SCADA, WSN, and RFID.</p>		
Case Study#3:	Industrial IoT (IIoT) Implementation using SCADA	
Case Study#4:	Retail Inventory Management using RFID	
Unit 3	IoT Data Processing Topologies	7 Hours
<p>Data Format- Structured data, Unstructured data, Importance of Processing in IoT, Processing Topologies- On-site processing, Off-site processing, IoT Device Design and Selection Considerations, Processing Offloading-Offload location, Offload decision making, Offloading considerations.</p>		
Case Study#5:	Autonomous Vehicle Data Processing (Edge/On-site)	
Case Study#6:	Smart Grid Energy Analytics (Cloud/Off-site)	
Unit 4	IoT Protocols	7 Hours
<p>Protocol Standardization for IoT, M2M, and WSN Protocols, RFID Protocol, Modbus Protocol, Zigbee Architecture. IP-based Protocols: MQTT (Secure), 6LoWPAN, LoRa.</p>		
Case Study#7:	Smart Grid Energy Analytics (Cloud/Off-site)	
Case Study#8:	Telemetry for Remote Pipelines using MQTT	
Unit 5	Cloud Platforms for IoT	8 Hours
<p>Software Defined Networking, Introduction to Cloud Storage Models, Communication API. WAMP: AutoBahn for IoT, Xively Cloud for IoT.</p> <p>Python Web Application Framework: Django Architecture and application development with Django, Amazon Web Services for IoT, SkyNet IoT Messaging Platform, RESTful Web Service, GRPC, SOAP.</p>		
Case Study#9:	Weather Monitoring Station with AWS IoT Core	
Case Study#10:	Building a Web-Dashboard for IoT using Django	
Unit 6	Security in IoT	8 Hours

Introduction, Vulnerabilities of IoT, Security Requirements, Challenges for Secure IoT, Threat Modelling.	
Key elements of IoT Security: Identity establishment, Access control, Data and message security, non-repudiation, and availability; security model for IoT; challenges in designing IoT applications; and lightweight cryptography.	
Case Study#11:	The Mirai Botnet Attack Analysis
Case Study#12:	Secure Medical Wearables (IoMT)
Total Hours: 45	
Text Books:	
<ol style="list-style-type: none"> 1. Arshdeep Bahga, Vijay Madiseti, "Internet of Things – A hands-on approach", Universities Press, ISBN: 0: 0996025510, 13: 978-0996025515 2. Olivier Hersent, David Boswarthick, Omar Elloumi, "The Internet of Things: Key Applications and Protocols", 2nd Edition, Wiley Publication, ISBN: 978-1-119-99435-0 3. Sudip Misra, Anandarup Mukherjee, and Arijit Roy. Introduction to IoT. Cambridge University Press, 2021. 4. Dow, Colin. Hands-On Edge Analytics with Azure IoT: Design and Develop IoT Applications with EdgeAnalytical Solutions, Including Azure IoT Edge. Packt Publishing Ltd, 2020. 	
Reference Books:	
<ol style="list-style-type: none"> 1. Dawoud Shenouda Dawoud, Peter Dawoud, "Microcontroller and Smart Home Networks", ISBN: 9788770221566, e-ISBN: 9788770221559 2. Charles Crowell, "IoT-Internet of Things for Beginners: An Easy-to-Understand Introduction to IoT", ISBN-13 : 979-8613100194 3. David Hanes, Gonzalo Salgueiro, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, ISBN-13: 978-1-58714-456-1 ISBN-10: 1-58714-456-5 4. Perros, Harry G. An Introduction to IoT Analytics. Chapman and Hall/CRC, 2021. 	
E-Books/Online Learning Resources:	
<ol style="list-style-type: none"> 1. https://www.iotforall.com/ebooks/an-introduction-to-iot 2. https://www.qorvo.com/design-hub/ebooks/internet-of-things-for-dummies 3. https://www.krishnagudi.com/wp-content/uploads/2020/04/Cisco-press-iot-Fundamentals.pdf 4. https://pg.its.edu.in/sites/default/files/KCA043%20Internet%20of%20things%20-IoT%20by%20Raj%20Kamal%20Text%20Book.pdf 	
Online Courses:	

1. Introduction to Internet of Things (2022)
 - a. Instructor: Prof. Sujoy Sinha Roy (IIT Kharagpur)
 - b. Covers IoT protocols (MQTT, CoAP), networking, constrained devices, Zigbee/802.15.4 basics
 - c. Link: https://onlinecourses.nptel.ac.in/noc22_cs53/preview
2. IoT Networks and Protocols
 - a. Instructor: Prof. Graeme Thomas (Curtin University)
 - b. Covers IoT data networks, protocols (MQTT/CoAP), layer models, fog/cloud integration
 - c. Link: <https://www.edx.org/learn/computer-networking/curtin-university-iot-networks-and-protocols>
3. Introduction to Internet of Things
 - a. Instructor: Arpan Pal (TCS Research)
 - b. Covers IoT communication protocols, REST/HTTP, sensor data protocols, BTech syllabus topics
 - c. Link: <https://www.coursera.org/learn/introduction-to-internet-of-things>

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3		-	-	-	-	-	-	-	-	1
CO2	2	2	3	-	2	-	-	-	-	-	-
CO3	3	1	-	-	-	-	-	-	-	-	-
CO4	-	3	-	1	-	-	-	-	-	-	-
CO5	-	-	3	-	3	-	-	-	-	-	1
CO6	-	-	3	-	-	3	1	-	-	-	-



Dnyaan Prasad Global University
Dr. D. Y. Patil Unitech Society
School of Technology and Research

Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
OE	Applied Machine Intelligence for FinTech Systems			Code	DP04020425405			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
3	3	--	--	45	--	30	--	75
Prerequisite:								
<ol style="list-style-type: none"> 1. Fundamentals of mathematics & statistics (linear algebra, probability) 2. Basic Python programming 								
Other Courses in this Series:								
<ol style="list-style-type: none"> 1. Python for Data Science 								
Course Objectives:								
<ol style="list-style-type: none"> 1. Understand core FinTech systems and financial data characteristics. 2. Apply machine learning techniques to solve FinTech problems. 3. Analyze and preprocess financial datasets for intelligent decision-making. 4. Build AI models for fraud detection, credit scoring, and risk analysis. 5. Evaluate model performance with appropriate metrics and ethical considerations. 6. Implement real-world FinTech use cases through hands-on projects. 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Understand financial datasets and their preprocessing requirements								
CO2: Apply ML models to core FinTech problems (Credit Risk & Financial Crime)								
CO3: Evaluate financial predictive models using standard metrics								
CO4: Design ML pipelines for transaction monitoring b, risk scoring, and customer analytics								
CO5: Apply responsible AI practices for financial decision-making								
Course Contents								
Unit 1	Foundations of AI & FinTech (6 Hrs.)							6 Hours
Introduction & Evolution of FinTech and AI adoption in financial services, FinTech in the context of Digital Economy (10 Building Blocks), Business functions: Payments, Lending & Crowdfunding,Regulatory landscape: Basel III / KYC/ AML								
Unit 2	Machine Learning Basics							8 Hours
Exploratory Data Analysis for finance, Intro to Machine Learning, Supervised vs. Unsupervised learning, Supervised learning: Regression models for financial forecasting , Supervised learning: Classification for fraud detection, loan approvals, Unsupervised learning: Clustering for customer segmentation, Ethical/Responsible AI in Finance								
Unit 3	Tools & Implementation							6Hours

Python, NumPy, Pandas, Scikit-Learn, Model evaluation: ROC-AUC, F1, precision/recall, KS-statistic, Recommendation systems for financial products (up-selling/ cross-selling)											
Unit 4	Financial Modeling with ML										8 Hours
Credit risk fundamentals: PD, LGD, EAD, Credit scoring models, Time-series modelling basics like trends, seasonality , Portfolio basics for stock/crypto/market trends, Anomaly detection in transaction patterns, Customer lifetime value prediction											
Unit 5	ML use cases in Banking										8 Hours
Robo-Advisory, WealthTech , RegTech (AML/KYC), InsurTech											
Unit 6	Mini Project										9 Hours
Develop models for: <ul style="list-style-type: none"> • Fraud detection • Credit risk scoring • Customer segmentation • Financial forecasting 											
											Total Hours: 45
Reference Books:											
1. Fintech: The New DNA of Financial Services, <i>Pranay Gupta, T. Mandy Tham</i> 2. Fintech and the Future of Finance Market and Policy Implications, <i>Erik Feyen, Harish Natarajan, and Matthew Saal</i>											
E-Books/Online Learning Resources:											
1. https://www.globallegalinsights.com/practice-areas/fintech-laws-and-regulations/india/ 2. https://en.wikipedia.org/wiki/Unified_Payments_Interface 3. https://www.google.com/search?q=fintech+journey+on+time&sca_esv=9c2282f08c1c1588&udm=2&biw=1920&bih=945&ei=Beo8ad2xJq2tseMP-_jLsAQ&ved=0ahUKewidyJHP27mRAxWtVmwGHXv8EkYQ4dUDCBI&uact=5&oq=fintech+journey+on+time&gs_lp=EgtnD3Mtd2l6LWltZyIXZmludGVjaCBqb3VybmV5IG9uIHRpbWVloBNQjgpYhxJwAngAkAEAmAFmoAGdBaoBAzYuMbgBA8gBAPgBAZgCAKACAjgDAIlgGAZIHAKAHVLIHALgHAMIHAMgHAIAIAQ&scient=gws-wiz-img#sv=CAMSXhoyKhBILUFyT2ZTZlozajhfbFFNMg5Bck9mU2ZaM2o4X2xRTToOallLOV94emg2UGpGWk0gBCokCg4yeV9VczhIOFRkR0U4TRIQZS1Bck9mU2ZaM2o4X2xRTRgAMAeyByCE7peECzACSGoIAhACGAEGaigB 4. https://www.researchgate.net/figure/A-timeline-showcasing-the-evolution-of-FinTech_fig1_388190350											
Online Courses:											
1. Coursera – AI in Finance Specialization 2. edX – FinTech: AI & Machine Learning Applications 3. Google Finance & Yahoo Finance APIs 4. Kaggle Financial Datasets 5. TensorFlow, PyTorch official tutorials											
The CO-PO Mapping Matrix											
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	—	2	2	—	—	—	—	—	1
CO2	3	3	2	2	3	—	—	—	—	—	1
CO3	2	3	—	3	2	—	—	—	—	—	1
CO4	3	2	3	2	3	1	—	2	1	2	1
CO5	1	2	1	—	—	3	3	1	1	—	2



Dnyaan Prasad Global University
Dr. D. Y. Patil Unitech Society
School of Technology and Research

Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
VSEC	WebX Design and Development Lab			Code	DP04020425406			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
2	--	--	4	--	30	--	20	50
Prerequisite:								
1) Problem Solving through Programming, Database Management Systems.								
Other Courses in this Series:								
1) Advanced Programming in WebX								
Course Objectives:								
1) To provide fundamental knowledge of web technologies and structured web development. 2) To enable students to develop dynamic and interactive web applications using modern frameworks. 3) To integrate backend services and databases for secure and scalable web applications. 4) To introduce deployment, security practices, and performance optimization in web applications.								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Develop structured and responsive web pages using HTML, CSS, and JavaScript.								
CO2: Build dynamic web applications using JavaScript frameworks and API integration.								
CO3: Implement backend integration and database connectivity for secure web applications.								
CO4: Deploy and manage web applications with proper configuration and optimized performance.								
Suggested List of Laboratory Experiments/Assignments								
Each batch must complete two assignments from group A,B,C and One assignment from Group D and E, ensuring that all assignments are collectively covered.								
Sr. No.	Group-A (Unit-1 Fundamentals of Web Technology)							
1	1) Online Library System – Develop a responsive webpage to display books description in table. 2) Event Registration System – Create an online form for seamless event registrations. 3) Personal Portfolio Website – Design a structured, mobile-friendly personal portfolio. 4) Feedback Form Page-Create a basic feedback form using HTML form elements and CSS styling.							

	Group-B (Unit-2 Interactive Web Development)
2	<ol style="list-style-type: none"> 1) Interactive Quiz App – Build a JavaScript-based quiz with real-time score updates. 2) Online application – Create a Gate registration form and validate using java script. 3) Smart To-Do List – Develop a dynamic to-do list with deadline reminders. 4) BMI Calculator App-Build an interactive BMI calculator using form inputs and JavaScript for logic and display.
	Group-C (Unit-3 Backend and API Integration)
3	<ol style="list-style-type: none"> 1) Student Attendance System – Implement an attendance report generator using React. 2) Smart Parking System – Develop a real-time web-based system to display available parking slots 3) Appointment Scheduling – Develop an online appointment booking system with notifications. 4) Online Food Ordering System- Develop a backend to handle menus, cart operations, and order processing.
	Group-D (Unit-4 Deployment & Future Web Technologies)
4	<ol style="list-style-type: none"> 1) E-commerce Deployment – Deploy an optimized online shopping platform. 2) Blogging Platform Deployment -Build and deploy a simple blog website. 3) Smart Health Tracker Dashboard-Host a health-monitoring dashboard.
	Group-E (Mini Project) (Form group 3-4 Students)
5	<ol style="list-style-type: none"> 1) Design and implement a dynamic web application for any business functionality by utilizing the web development technologies covered in the above assignments. The project should incorporate frontend design, interactive features, backend integration, and deployment to provide a complete web-based solution. 2) Online Voting System for College Elections: Implement a secure online voting portal for class representative or student council elections. Admin can register voters and candidates, and students can cast one vote per election. Security: Email OTP or login-based verification 3) College Internship and Placement Portal: Create a platform for students to apply for internships or placement drives. Admins (TPO) can post opportunities, and companies can review applications. Few features are Student dashboard, application tracker, resume upload, admin panel.
Textbooks:	
<ol style="list-style-type: none"> 1. Jeffrey Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035 2. Jon Duckett, "HTML & CSS: Design and Build Websites", 1st Edition, Wiley, ISBN: 978-1118008188. 3. Marijn Haverbeke, "Eloquent JavaScript", 3rd Edition, No Starch Press, ISBN: 978-1593279509 	
Reference Books:	
<ol style="list-style-type: none"> 1. Author TBD, "Modern Web Development with WebX", 1st Edition, Publisher TBD, ISBN: TBD 2. Mario Casciaro, "Node.js Design Patterns", 3rd Edition, Packt Publishing, ISBN: 978-1839214110 3. Mark Masse, "REST API Design Rulebook", 1st Edition, O'Reilly Media, ISBN: 978-1449310509 4. Adam Freeman, "Pro Angular", 5th Edition, Apress, ISBN: 978-1484264489 	
E-Books/Online Learning Resources:	

1. HTML & CSS by Jon Dockett,” Visual approach to learning web design basics”,<https://bcs.wiley.com/he-bcs/Books?action=mininav&bcsId=6056&itemId=1118008189&assetId=283213>.
2. Fullstack React ,” Hands-on guide to building apps using React + Redux + APIs”,
<https://www.fullstackreact.com/>.
3. <https://www.w3.org/html/>.
4. [HTML, The Complete Reference http://www.htmlref.com/](http://www.htmlref.com/).
5. <http://www.tutorialspoint.com/>.
6. <https://www.geeksforgeeks.org/web-technology/>.

Online Courses:**1. Web Technology**

- a. Instructor: Dr. Ashutosh Kumar Bhatt.
- b. Covers Web Technology Fundamentals, HTML, CSS, JavaScript, JSP.
- c. Link: https://onlinecourses.swayam2.ac.in/nou24_cs09/preview. 2

Introduction to Modern Application Development, IIT Madras

- a. Instructor: By Prof. Gaurav Raina, Tanmai Gopal.
- b. Covers development of single-page web applications, dependency injection, typescript, components, and directives with this Angular Course.
- c. Link: <https://nptel.ac.in/courses/106106156> .

Software Tools, Simulation Platforms & Hardware Kits:

1. Visual Studio Code – Open-source code editor with extensions for web development.
<https://code.visualstudio.com/>.
2. Google Chrome Developer Tools – Built-in browser tools for debugging and performance analysis. <https://developer.chrome.com/docs/devtools/>.
3. Postman – API development and testing tool. <https://www.postman.com/>.
4. Node.js & npm – JavaScript runtime and package manager for backend development. <https://nodejs.org/>.
5. Firebase – Cloud platform for hosting and backend services.
<https://firebase.google.com/>.
6. MongoDB Atlas – Cloud-based NoSQL database service.
<https://www.mongodb.com/cloud/atlas>
7. MySQL Workbench – SQL database management tool.
<https://www.mysql.com/products/workbench/>.
8. Figma – UI/UX design and prototyping tool. <https://www.figma.com/>.
9. GoogleSites-Simple free site builder (good for internal sites)- sites.google.com
10. CodePen-Test & share frontend code (HTML/CSS/JS)-codepen.io

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	–	3	1	–	–	2	–	1
CO2	3	2	3	–	3	1	–	1	2	–	1
CO3	3	3	2	2	3	1	2	–	2	1	2
CO4	2	2	2	2	3	1	–	1	2	2	2



Dnyaan Prasad Global University
Dr. D. Y. Patil Unitech Society
School of Technology and Research

Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
PCC	Computer Network and Security Lab			Code	DP04020425407			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
1	--	--	2	--	--	--	25	25
Prerequisite:								
1) Basics of Computers								
Other Courses in this Series:								
1) Computer Network and Security.								
Course Objectives:								
<ol style="list-style-type: none"> To provide hands-on understanding of computer network fundamentals and layered architecture. To implement and analyze data link, network, transport, and application layer protocols. To study network performance, routing, and addressing mechanisms. To gain practical exposure to network security threats, cryptographic techniques, and secure protocols. 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Demonstrate the working of basic computer networking concepts and layered models.								
CO2: Implement and analyze data link layer protocols and LAN technologies.								
CO3: Apply IP addressing and routing concepts in network design.								
CO4: Analyze transport layer protocols and congestion control mechanisms.								
CO5: Demonstrate the working of application layer protocols using network analyzers.								
CO6: Identify security threats and apply cryptographic and security protocols for secure communication.								
Suggested List of Laboratory Experiments/Assignments Assignments from all Groups (A, B, C, D, E, and F) are compulsory								
Sr. No.	Group A: Introduction to Computer Networks (Unit-1) (Attempt any two)							
	<ol style="list-style-type: none"> To learn handling and configuration of networking hardware like RJ-45 connectors, CAT-6 cable, crimping tool, etc. Study of network topologies and components in a small office/campus network. Bandwidth and latency measurement for different network connections. 							

	Group B: Data Link Layer (Unit-2) (Attempt any two)
	<ol style="list-style-type: none"> 1. Implementation of Stop-and-Wait Protocol and Sliding Window Protocol. 2. Simulation of framing, error detection, and correction techniques. 3. Study of MAC protocols (ALOHA, CSMA/CD, CSMA/CA) using simulation tools. 4. Study and configuration of Ethernet and Wireless LAN standards. 5. Code simulating ARP / RARP protocols. 6. Network packet analysis using tools like Wireshark, tcpdump (Frame-level analysis).
	Group-C: Network Layer (Unit-3) (Attempt any two)
	<ol style="list-style-type: none"> 1. Commands/services involving network layer ping, traceroute, ARP, and nslookup. 2. Implementation of IPv4 addressing, subnetting, and supernetting. 3. Study and simulation of routing algorithms (Distance Vector and Link State). 4. Comparative study of IPv4 and IPv6 packet formats.
	Group-D: Transport Layer (Unit-4) (Attempt any one)
	<ol style="list-style-type: none"> 1. Simulation of flow control and congestion control mechanisms. 2. Comparative study of TCP, UDP, and QUIC protocols.
	Group-E: Application Layer (Unit-5) (Attempt any one)
	<ol style="list-style-type: none"> 1. Analysis of application layer protocols (HTTP, FTP, SMTP, DNS) using Wireshark. 2. Study of DHCP configuration and its impact on enterprise networks.
	Group-F: Network Security (Unit-6) (Attempt any one)
	<ol style="list-style-type: none"> 1. Study and analysis of security protocols (TLS, HTTPS, SSH, IPsec) using a protocol analyzer. 2. Demonstration of cryptographic techniques for secure communication.
Text Books:	
<ol style="list-style-type: none"> 1. Behrouz a. Forouzan, "Data Communications and Networking", 5th Edition, McGraw Hill Education, ISBN: 978-0-07-337622-6 2. Andrew Tanenbaum, "Computer Networks", 5th Edition, Pearson, ISBN-13: 978-0132126953 ISBN-10: 0132126958. 3. Forouzan. TCP/IP Protocol Suite. 4th Edition. 4th Edition - 27 March 2009. ISBN-13: 978-0073376042 ISBN-10: 0073376043. 	
Reference Books:	
<ol style="list-style-type: none"> 1. James Kurose, Keith Ross, "Computer Networking: A Top-Down Approach", 7th Edition, Pearson Education, ISBN: 978-0133594140. 2. William Stallings, "Data and Computer Communications", 4th Edition, McGraw Hill, ISBN: 978-0070411838. 	
E-Books/Online Learning Resources:	
<ol style="list-style-type: none"> 1. Computer Networking: Principles, Protocols and Practice, fourth edition https://4ed.computer-networking.info/syllabus/default/index.html 2. CCNA: Introduction to Networks, https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US 3. Computer Networks: A Systems Approach https://open.umn.edu/opentextbooks/Textbooks/771 4. Computer Network Tutorial https://www.geeksforgeeks.org/computer-networks/computer-network-tutorials/ 	

Online Courses:

1. Computer Networks and Internet Protocol

A. **Instructor:** Prof. Ajit Pal (IIT Kharagpur)B. **Covers:** Fundamentals of computer networks, TCP/IP protocol stack, routing, congestion control, socket programming, SDN, and network security.C. **Link:** <https://nptel.ac.in/courses/106105080>

2. Computer Networks

A. **Instructor:** Prof. Sandip Chakraborty & Prof. Soumya Kanti Ghosh (IIT Kharagpur)B. **Covers:** Fundamentals of computer networks, TCP/IP protocol stack, routing, congestion control, socket programming, SDN, and network security.C. **Link:** https://onlinecourses.nptel.ac.in/noc25_cs15/preview**Software Tools, Simulation Platforms& Hardware Kits:**

Wireshark

Cisco Packet Tracer / NS-3

Linux Networking Tools

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	-	-	-	-	-	-	-	-	-
CO2	3	2	2	-	1	-	-	-	-	-	-
CO3	3	2	2	1	-	-	-	-	-	-	-
CO4	3	3	2	1	-	-	-	-	-	-	-
CO5	2	2	-	1	2	-	-	-	-	-	-
CO6	3	2	-	-	2	1	-	2	-	-	2



Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
AEC	Effective Technical Communication			Code	DP04020425408			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
2	2	--	--	50	--	--	--	50
Prerequisite: NA								
Other Courses in this Series: NIL								
Course Objectives: <ol style="list-style-type: none"> 1. Develop clarity, accuracy, conciseness, and professionalism in technical and scientific communication across written, oral, and digital formats. 2. Apply the scientific method and structured writing processes (planning, drafting, revising, editing, proofreading) to improve communication effectiveness in engineering contexts. 3. Understand linguistic and communicative competence, including language structures, registers, and cross-cultural communication, for collaborative technical work. 4. Analyze and select appropriate modes and media of technical communication, including reports, presentations, posters, patents, and web-based content. 5. Demonstrate ethical, audience-centered, and accessible communication practices while documenting and presenting engineering projects and research outcomes. 								

<p>Course Outcomes:</p> <p>On completion of the course, students will be able to:</p> <p>CO1: Apply principles of clarity, honesty, accuracy, conciseness, and accessibility to produce effective technical documents and presentations.</p> <p>CO2: Use the scientific method and writing process to diagnose communication problems and systematically improve technical communication in team-based engineering projects.</p> <p>CO3: Analyze linguistic foundations (phonology, morphology, syntax, semantics, pragmatics) and adapt language registers for effective spoken and written communication in diverse technical teams.</p> <p>CO4: Compare, design, and produce appropriate technical communication modes, including reports, posters, manuals, patents, SOPs, and web-based technical content using digital tools and platforms.</p> <p>CO5: Present and document engineering projects ethically and professionally, demonstrating responsible authorship, data integrity, audience awareness, and compliance with documentation standards.</p>		
Course Contents		
Unit 1		6 Hours
<p>Technical Communication:</p> <p>Learning goals: Emphasis on clarity, honesty, accuracy, conciseness, accessibility, professionalism</p> <p>Writing Process: Focus on planning, drafting, revising, editing, and proofreading</p> <p>Understanding the scientific method in communication</p>		
Case Study#1:	<p>Applying the Scientific Method to Improve Technical Communication in a Design Team</p> <p>Scenario:</p> <p>An engineering team developing a smart irrigation system faces miscommunication between hardware, software, and data teams, causing delays and mismatches. To address this, their faculty advisor asks them to apply the scientific method to identify and improve their communication process. (Role-play Activity, Team Presentation etc)</p>	
Case Study#2:	<p>Enhancing Professional Communication in an Engineering Project by Communicating Failure Analysis Results.</p> <p>Scenario:</p> <p>Engineering students develops a low-cost water purification system but faces a failure in 3D-printed components. Their initial technical report and presentation are unclear, overly technical, and lack structure. The mentor asks them to revise their work, focusing on key principles</p>	
Unit 2		6 Hours
<p>Foundations of Communicative and Linguistic Competence:</p> <p>Structure and functions of language, Language acquisition vs. language learning,</p> <p>Registers and varieties of language: formal-informal, technical-general</p> <p>Linguistic Foundations: phonology, morphology, syntax, semantics, and pragmatics</p>		

Case Study#3:	Analyse the linguistic foundations of effective spoken communication on the student's team working on a campus robotics competition consists of students from different linguistic backgrounds. (e.g. Identify Communication Gaps, Apply Linguistic Strategies, Group Reflection etc)
Unit 3	6 Hours
Modes of Technical Communication: Comparative analysis of technical documents: Reports, Papers, Manuals, Book Chapters, Abstracts & Synopses Poster design using tools: Canva, PowerPoint Patent Drafting and IP Documents: Grant Applications and Statement of Purpose (SOP)	
Case Study#4:	Case Study: "Choosing the Right Mode: Communicating Technical Solutions in a Real-World Engineering Project"
Unit 4	6 Hours
Web-based Technical Content: Introduction to Web-based Technical Writing, Types of Web-based Technical Content: FAQs, help docs, knowledge base articles, blogs, white papers, tutorials Content Management Systems (CMS): WordPress, GitHub Pages Visual & Multimedia Integration	
Case Study#5:	<p>Students have developed a Python-based chatbot using Natural Language Processing (NLP). They now wish to share their experience and knowledge by publishing a web-based technical blog targeted at junior students, peers, and open-source enthusiasts.</p> <p>Discuss the case study: From Code to Content- Writing and Publishing a Technical Blog for Real-World Learners</p>
Unit 5	6 Hours
Technical Presentation and Documentation Skills: Principles of Effective Technical Presentations: Audience analysis, slide design, delivery techniques Scientific Documentation: Writing lab reports, project documentation, compliance and documentation standards Ethics in Technical Communication: Plagiarism, data integrity, responsible authorship	
Case Study#6:	<p>A group of students has created a simple mobile app that helps students find free classrooms on campus. Their project is selected for a college tech fest, where they need to present it to fellow students, teachers, and visitors.</p> <p>Discuss - how students will explain technical ideas clearly and confidently using presentations, posters, and short descriptions.</p>
Total Hours: 30	
Textbooks:	
<ol style="list-style-type: none"> 1. M. Ashraf Rizvi – Covers all four language skills (listening, speaking, reading, writing) with clear pedagogy and practice-oriented modules for engineers. 2. Meenakshi Raman & Sangeeta Sharma – A comprehensive text with extensive examples and exercises tailored to undergraduate engineering learners. 	
Reference Books:	

1. Gajendra Singh Chauhan & Smita Kashiramka

Well-structured text covering grammar, writing, presentations, and soft skills.

2. K. Ramesh Rao A concise reference with clear concept explanations and exercises.

E-Books/Online Learning Resources:

BCcampus OpenEd – Free textbooks on communication skills

Saylor Academy – Free modules including professional communication

Online Courses:

1. NPTEL – Technical Communication

2. Coursera – Technical Writing

3. edX – Professional Communication

The CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	1	-	-	1	-	2	-	3	-	1
CO2	2	3	-	2	1	-	1	2	3	-	2
CO3	-	1	-	-	-	1	1	3	3	-	2
CO4	1	1	2	-	3	-	1	2	3	1	2
CO5	-	1	1	1	-	2	3	2	3	1	2



Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
E/E/M	Applied Machine Intelligence for FinTech Systems Lab			Code	DP04020425409			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
2	--	--	4	--	50	--	--	50

Prerequisite:

1. Fundamentals of mathematics & statistics (linear algebra, probability)
Basic Python programming

Other Courses in this Series:**Course Objectives:**

1. Apply supervised, unsupervised, and time-series machine learning techniques to solve real-world financial problems such as credit scoring, forecasting, fraud detection, and customer segmentation.
2. Design, train, tune, and evaluate machine learning models using appropriate datasets, metrics, and validation strategies relevant to financial decision-making.
3. Handle practical data challenges in finance, including class imbalance, non-stationarity, noisy data, and operational constraints.
4. Interpret, explain, and assess machine learning models with respect to calibration, fairness, and ethical considerations in financial applications.
5. Build end-to-end machine learning pipelines and deploy lightweight predictive services, demonstrating professional documentation and teamwork.

Course Outcomes:

CO1: Develop and evaluate **supervised learning models** for financial classification and regression problems using appropriate metrics, validation methods, and comparison techniques.

CO2: Analyze financial datasets to identify issues such as **class imbalance, non-stationarity, and anomaly patterns**, and apply suitable preprocessing and modelling strategies.

CO3: Design and implement **unsupervised and semi-supervised models** for tasks such as fraud detection and customer segmentation, and justify model and parameter choices.

CO4: Interpret and evaluate machine learning models using **explainability, calibration, and fairness techniques**, and propose mitigation strategies for ethical and operational risks.

CO5: Build a **complete ML workflow**, including data pipelines, model persistence, API-based deployment, and documentation, demonstrating readiness for real-world financial ML applications.

Suggested List of Laboratory Experiments/Assignments

Each batch must complete two assignments from group A,B,C and One assignment from Group D and E, ensuring that all assignments are collectively covered.

Sr. No.	Lab1- Supervised Learning: Credit Scoring (Classification)
1	<p>Objective: Build and evaluate credit default (binary) classification models. Datasets: UCI Credit Card Default dataset, Lending Club (processed), or synthetic credit dataset. Tasks:</p> <ol style="list-style-type: none"> 1. Split data with time- or stratified-sampling. 2. Baseline logistic regression; regularization and feature selection. 3. Try tree-based models (Random Forest, XGBoost if allowed). 4. Handle class imbalance (SMOTE, class weights). 5. Evaluate using ROC-AUC, precision-recall, confusion matrix, calibration plots. <p>Expected outcome: Model notebook with comparisons and recommended model and rationale.</p>
	Lab2- Regression for Financial Forecasting (Time-series basics)
2	<p>Objective: Short-term forecasting using classical and ML regression approaches. Datasets: Stock/ETF price series (Yahoo Finance), daily exchange rates, or synthetic series. Tasks:</p> <ol style="list-style-type: none"> 1. Stationarity checks (ADF), differencing if needed. 2. Fit ARIMA/SARIMAX baseline. 3. Build a supervised ML regression baseline (lag features + tree/linear models). <p>Backtesting with rolling-window cross-validation; compute MAPE, RMSE.</p> <p>Expected outcome: Forecasting notebook showing model comparisons and best-performing approach for horizon selected.</p>
	Lab3- Fraud Detection: Anomaly Detection & Classification

3	<p>Objective: Detect fraudulent transactions using unsupervised and supervised approaches.</p> <p>Datasets: Kaggle Credit Card Fraud dataset or synthetic imbalanced transaction dataset.</p> <p>Tasks:</p> <ol style="list-style-type: none"> 1. Baseline unsupervised detectors (IsolationForest, OneClassSVM), evaluate via precision@k. 2. Supervised approach (if labels available): stacked sampling, threshold calibration. 3. Build simple autoencoder for anomaly scoring (optional). 4. Operational considerations: latency, false positive cost analysis. <p>Expected outcome: Working detectors, comparison table, discussion of deployment trade-offs.</p> <p>Assessment (15 pts): Detector performance & evaluation (7), reasoning about cost of errors & operationalization (5), code quality (3).</p>
Lab-4 Customer Segmentation & CLV Estimation (Unsupervised + Regression)	
4	<p>Objective: Segment customers and estimate Customer Lifetime Value (CLV).</p> <p>Datasets: Transaction history with customer IDs and timestamps.</p> <p>Tasks:</p> <ol style="list-style-type: none"> 1. RFM analysis & feature normalization. 2. Cluster customers (KMeans / GMM), choose K with silhouette/Elbow. 3. Estimate CLV using simple margin \times expected transactions (or BG/NBD if time permits). 4. Visualize segments and recommend business actions per segment. <p>Expected outcome: Segmentation report, CLV ranking, marketing suggestions.</p> <p>Assessment (10 pts): Segmentation quality & justification (4), CLV computation correctness (3), business insights (3).</p>
Lab 5 — Model Evaluation, Explainability & Fairness in Finance	
5	<p>Objective: Understand model interpretability (SHAP/LIME), calibration, and fairness checks for financial models.</p> <p>Datasets: Any model outputs from earlier labs (credit scoring).</p> <p>Tasks:</p> <ol style="list-style-type: none"> 1. Produce SHAP value explanations for top model predictions. 2. Test calibration & produce calibration plot; apply isotonic/Platt scaling. 3. Run fairness checks across protected groups (e.g., gender/age proxies) and compute disparate impact/metrics. 4. Document mitigation strategies (reweighing, post-processing). <p>Expected outcome: Explainability notebook and fairness audit with recommendations.</p> <p>Assessment (12 pts): Depth of explainability analysis (5), fairness checks & mitigation proposal (5), documentation clarity (2).</p>
Lab 6 — End-to-End ML Pipeline & Lightweight Deployment	
6	<p>Objective: Build a simple end-to-end pipeline: data ingestion \rightarrow model \rightarrow REST API serving.</p> <p>Datasets: Use a trained model from prior lab (e.g., credit scoring).</p> <p>Tasks:</p> <ol style="list-style-type: none"> 1. Create data preprocessing pipeline (fit/transform serialized). 2. Save model + pipeline (joblib). 3. Expose a REST endpoint for prediction (Flask/FastAPI) and test with curl/postman. <p>Discuss CI/CD & monitoring basics (brief).</p> <p>Expected outcome: Working local API returning predictions and input validation.</p>
Lab 7 — Mini Project: Choose & Implement (Capstone for Sem 4)	

7	<p>Objective: Apply learned techniques to a small end-to-end FinTech problem. Team of 2–3 students.</p> <p>Suggested topics: Fraud detection, credit scoring, churn prediction, transaction anomaly monitoring, small robo-advisor prototype.</p> <p>Deliverables: Code repo, final report (6–8 pages), short demo (5–10 mins).</p>										
Textbooks:											
1. Fintech: The New DNA of Financial Services, <i>Pranay Gupta, T. Mandy Tham</i> 2. Fintech and the Future of Finance Market and Policy Implications, <i>Erik Feyen, Harish Natarajan, and Matthew Saal</i>											
Reference Books:											
E-Books/Online Learning Resources:											
1. https://www.globallegalinsights.com/practice-areas/fintech-laws-and-regulations/india/ 2. https://en.wikipedia.org/wiki/Unified_Payments_Interface 3. https://www.google.com/search?q=fintech+journey+on+time&sca_esv=9c2282f08c1c1588&udm=2&biw=1920&bih=945&ei=Beo8ad2xJq2tseMP-jLsAQ&ved=0ahUKEwidyJHP27mRAxWtVmwGHXv8EKYQ4dUDCBI&uact=5&oq=fintech+journey+on+time&gs_lp=Egtnd3Mtd2l6LWltZyIXZmludGVjaCBqb3VybmV5IG9uIHRpbWVloBNQjgpYhxJwAngAkAEAmAFmoAGdBaoBAzYuMbgBA8gBAPgBAZgCAKACAjgDAIqGAZIHAKAHVLIHALgHAMIHAMgHAIAlAQ&sclient=gws-wiz-img#sv=CAMSXhoyKhBILUFyT2ZTZlozajhfbFFNMq5Bck9mU2ZaM2o4X2xRTToOallOV94emg2UGpGWk0gBCokCg4yeV9VczhIOFRkR0U4TRIQZS1Bck9mU2ZaM2o4X2xRTgAMAeyByCE7peECzACSGolAhACGAEGAgigB 4. https://www.researchgate.net/figure/A-timeline-showcasing-the-evolution-of-FinTech_fig1_388190350											
Online Courses:											
Coursera – AI in Finance Specialization • edX – FinTech: AI & Machine Learning Applications • Google Finance & Yahoo Finance APIs • Kaggle Financial Datasets											
Software Tools, Simulation Platforms & Hardware Kits:											
scikit-learn (KMeans/GMM), lifetimes package (BG/NBD) or custom RFM, pandas, imbalanced-learn, matplotlib, joblib, Flask or FastAPI, Docker (optional), shap, lime, scikit-learn, calibration_curve, IsolationForest, autoencoders (simple), imbalanced-learn.											
The CO-PO Mapping Matrix											
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	1	3	-	-	1	1	-	2
CO2	2	3	-	3	2	-	-	-	-	-	2
CO3	2	3	3	2	3	1		1	-	-	2
CO4	1	2	1	2	2	2	3	-	1	-	2
CO5	2	2	3	1	3	-	1	2	2	2	3



Dnyaan Prasad Global University
Dr. D. Y. Patil Unitech Society
School of Technology and Research

Program	B.Tech. (Computer Science & Engineering)			Semester	IV			
VEC	Constitution of India			Code	DP04020425410			
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
2	2	--	--	30	--	20	--	50
Prerequisite:								
1) Social Studies								
Other Courses in this Series: Nil								
Course Objectives:								
<ol style="list-style-type: none"> To introduce the students to the history, philosophy, and structure of the Indian Constitution. To help students understand their rights and responsibilities as citizens. To provide awareness of governance, legal, and constitutional frameworks that influence engineering practice. To instil a sense of ethical and civic duty. 								
Course Outcomes:								
On completion of the course, students will be able to:								
CO1: Understand the philosophy, structure, and functioning of the Indian Constitution.								
CO2: Explore and value the importance of fundamental rights and duties.								
CO3: Interpret Role of Judiciary and Public Interest Litigation								
CO4: Identify the roles and responsibilities of key constitutional bodies.								
CO5: Apply constitutional principles to engineering and societal contexts.								
Course Contents								
Unit 1	Historical Background & Making of the Constitution						4 Hours	
Introduction to the Constitution of India and its salient features, Constituent Assembly and Drafting Committee, Preamble and its significance								
Case Study#1:	Comparative study of UK Parliamentary System vs Indian Parliamentary System							
Unit 2	Fundamental Rights & Fundamental Duties						8 Hours	
Fundamental Rights (Articles 12–35), Directive Principles of State Policy, Fundamental Duties (Article 51A)								
Case Study#2:	Justice K.S. Puttaswamy (Retd.) & Anr. vs. Union of India & Ors (2017 Privacy Judgment)							

Unit 3	Union and State Governments										5 Hours
Union Executives, State Executives, Role of Judiciary and Public Interest Litigation											
Unit 4	Emergency Provisions and its impact										5 Hours
Emergency Provisions (National, State, Financial)											
Case Study#3:	Basic Structure Doctrine (Kesavananda Bharati Sripadagalvaru and Ors vs. State of Kerala and Anr										
Unit 5	Constitution and Engineering Practice										8 Hours
Legal awareness in professional practice, Environmental safety, Labor Rights, RTI Act , Information Technology Act, 2000 , Personal Data Protection Bill (PDPB) / Digital Personal Data Protection Act, 2023 , National Cyber Security Policy, 2013, The Companies Act, 2013, CERT-In (Computer Emergency Response Team - India) Guidelines (For all acts only provisions study is expected)											
Case Study#4	Role and Responsibilities of National Green Tribunal (NGT)										
Total Hours: 20											
Text Books:											
1. Basu, D. D. (2013), " <i>Introduction to the Constitution of India</i> ", 21 st Edition, LexisNexis, ISBN: 9788180389185											
2. Laxmikanth, M. (2021), " <i>Indian Polity</i> ", 7 th ed.). McGraw Hill Education, ISBN: 9789354600353											
3. Rai, Udai Raj(2022), " <i>Constitutional Law: Governance Structure</i> ", 2 nd Edition, Eastern Book Company, ISBN: 9789390715260											
Reference Books:											
1. Government of India. (2023). <i>The Constitution of India (Bare Act)</i> . New Delhi: Government of India Press											
E-Books/Online Learning Resources:											
1. Ministry of Law and Justice, Government of India. (2023). <i>The Constitution of India</i> [E-book]. Retrieved from https://legislative.gov.in/constitution-of-india											
Online Courses:											
1. NPTEL: Advanced Constitutional Law											
a. Instructor: Dr. Anupama Goel (National Law University Delhi)											
b. This course will explore the ways in which courts, governments and individuals shape Indian constitutional law.											
c. Link: https://onlinecourses.swayam2.ac.in/cec20_hs22/preview											
2. NPTEL: Constitution Law and Public Administration in India											
a. Instructor: Prof. Sairam Bhat (National Law University Delhi)											
b. This course will explore wo major branches of law that govern the functioning of state and administrative agencies are: (i) Constitutional Law and (ii) Administrative law											
c. Link: https://onlinecourses.nptel.ac.in/noc24_lw05/preview											
The CO-PO Mapping Matrix											
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	—	1	—	—	—	3	2	—	—	—	1
CO2	—	1	—	—	—	3	3	1	—	—	1
CO3	—	2	—	1	—	3	3	—	1	—	1
CO4	—	1	—	—	—	3	2	—	—	—	1
CO5	1	2	2	—	—	3	3	1	2	1	2



Program	B.Tech. (Computer Science & Engineering)			Semester		IV		
MNC	Emotional Well Being			Code		DP04020425411		
	Teaching Scheme (Hrs/Week)			Examination Scheme & Marks				
Credits	Lectures	Tutorials	Practical	CA		TEA		Total
				Theory	Practical	Theory	Practical	
1	2	--	--	--	--	--	--	--
Prerequisite: Nil								
Other Courses in this Series:								
<ol style="list-style-type: none"> 1. Setting boundaries 2. Developing self-awareness 3. Developing personal and professional self. 								
Course Objectives:								
<ol style="list-style-type: none"> 1. Understand the basic concepts of psychology and mental health, including the roles of mental health professionals and the relevance of psychological wellbeing in academic and personal life. 2. Develop effective study habits, goal-setting, decision-making, and self-reflection skills to enhance academic performance and personal growth. 3. Identify and manage life transitions and emotional challenges by developing self-awareness, coping strategies, and resilience during periods of change. 4. Recognize, understand, and manage stress, anxiety, fear, and phobias using psychological techniques such as mindfulness, relaxation, and cognitive strategies. 5. Enhance communication skills, including clarity, assertiveness, boundary setting, and conflict resolution, to improve interpersonal effectiveness. 6. Apply psychological principles in real-life situations through reflective practices, group activities, discussions, and experiential learning. 								

Course Outcomes:

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

- CO1:** Synthesize Psychological Theories with Real-World Academic and Career Challenges.
CO2: Create Personalized Transition Plans for Complex Life and Academic Shifts:
CO3: Cultivate Deep Empathy and Assertiveness for Leadership in Diverse, High-Stakes Contexts.
CO4: Foster Emotional Intelligence for Leadership in a Globalized, High-Pressure Academic World.
CO5: Master Advanced Resilience Techniques for Unpredictable Career and Academic Environments.
CO6: Utilize Psychological Insights to Navigate Ethical Dilemmas in Professional and Academic Settings.
CO7: Develop an understanding of rationalization of thought in daily functioning and coping with intense challenges.

Course Contents

Unit 1	Introduction to Psychology, and Counselling	2 Hours
Importance and relevance in Academics: Definition of Psychology and Mental Health, Mental Health Professionals and their roles, study habits, goal setting and decision making in academics.		
Case Study#2:	Activity - Goal setting	
Unit 2	Transitioning Through Changes	3 Hours
Handling life transitions: Group Activity - The open mind challenge. Identify and understand the different types of life transitions (academic, professional, personal), Develop emotional self-awareness during periods of change, Cycle of Change , Coping strategies.		
Case Study#1:	Journal your respective life challenges in your diary and discuss how it has impacted you	
Case Study#2:	Interview someone who has gone through a trauma. (Home assignment)	
Unit 3	Resilience: Strategies to cope with setbacks and build a growth mindset	3 Hours
Understanding Stress, Types and its impact on our daily lives -Physiological and psychological impact, Growth Mindset: Principles and practices. (Fixed mindset vs Growth mindset), Ways to build resilience (cognitive flexibility, and developing supportive network)		
Case Study#1:	Activity: Exploring Resilience : Propose a situation and encourage responses from students.	
Unit 4	Effective Communication: Developing clarity and assertiveness.	2 Hours
Communication, Definition and Types, Communication Gaps: Common barriers and their resolution, Activity: Techniques to effectively articulate and communicate. (demo), Boundaries and Assertiveness, Activity - "Misunderstanding !		
Unit 5	Understanding and Managing Anxiety	2 Hours
Understanding Anxiety, Fear symptoms, and overcoming anxiety. Understanding Phobia		
Case Study#1:	Reflective activity 1 - penning down the nature of thoughts and identifying an anxious thought.)	
Case Study#2:	Activity 2 : Mindful breathing technique, progressive muscle relaxation, things in and out of control.	
Unit 6	Group Activities	3 Hours
Real-life applications of psychological skills / Psychological movie screening / debates.		
Total Hours: 15		
Text Books:		
1) Psychology: Themes and Variations by Wayne Weiten		
2) Understanding Psychology by Robert S. Feldman		

Reference Books:											
1) Counselling Skills and Theory by Margaret Hough											
E-Books/Online Learning Resources:											
1. https://www.apa.org/news/podcasts/speaking-of-psychology/health-anxiety											
2. https://www.helpguide.org/mental-health/stress/stress-management											
3. https://www.apa.org/topics/stress/tips											
4. https://psylearners.psychotechservices.com/2014/10/psychology-5th-edition-by-robert-baron_29.html											
5. https://books.google.co.in/books?id=PVQDAedulXcC&printsec=copyright&redir_esc=y#v=onepage&q&f=false (B.K. Mishra)											
6. https://www.google.co.in/books/edition/Psychology/TjFpDwAAQBAJ?hl=en&gbpv=1 (Ciccarelli)											
Online Courses:											
1. NPTEL: Mental Health and Wellbeing											
a. Instructor: Prof. Santanu Misra, Dr. Oyin Mibang and others (IIT Kanpur)											
b. The course provides practical exercises, mindfulness techniques, and skill-building modules focused on conflict resolution, time management, and goal setting, tailored for students navigating the challenges of college life.											
c. Link: https://onlinecourses.nptel.ac.in/noc25_hs109/preview											
The CO-PO Mapping Matrix											
CO/PO	PO1	PO 2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	3	2	2	1	2	1	2	2	1	3
CO2	1	2	2	1	1	2	1	2	2	2	3
CO3	—	1	1	—	—	3	3	3	3	2	2
CO4	—	1	1	—	—	3	3	3	3	2	2
CO5	—	2	1	1	—	2	1	2	1	2	3