Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

NEP Compliant Structure & Syllabus

of

Department of Computer Science & Engineering (Internet of Things & Cyber Security Including Block Chain Technology)

Pattern 'A-24'

S. Y. B. Tech.

Effective from Academic Year 2025-26

Prepared by: - Board of Studies of CSE-IoTCSBT

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Chairman- BoS Dean - Academics Chairman – Academic Board

Contents

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

Academic Information – Please visit www.vit.edu

FF No.: 653

Structure

Branch: CSE-IoTCSBT Year: Second Year (SY) Academic Year 2025-26

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week))			Total	Credits
			Theory	Lab	Tut		
S1	CB2003	Fundamentals of Data Structures	3	2	0	5	4
S2	CB2004	Data Communication and Networking	2	0	1	3	3
S3	CB2005	Basics of IoT	2	2	0	4	3
S4	CB2006	Object Oriented Programming	1	2	0	3	2
MDM	CBM001	Internet of Things	2	0	1	3	3
HSS	HS2002	From Campus to Corporate - 1	2	0	0	2	2
RAD	HS2001	Reasoning and Aptitude Development - 3	1	0	0	1	1
DT	CB2001	Design Thinking – 1	0	0	1	1	1
ISMA	IR2101	OR International Tech Front Module	-	-	-	-	1
EDI	CB2002	Engineering Design and Innovation – 1	0	4	0	4	2
IEDP	IR2102	OR International Innovation Program	-	-	-	-	2
Total			**13**	**10**	**3**	**26**	**21**

Examination scheme

Subject No.	Subject Code	Subject Name	Examination scheme										Total	Credits
			CVV	CP	LAB CA	GD/ PPT/ HA	MSE (W/O)	T1 (W/O)	T2 (W/O)	ESE	ESE (W/O)	PRACT /CVV		
S1	CB2003	Fundamentals of Data Structures	20	30	10						40(W)		100	4
S2	CB2004	Data Communication and Networking	30			GD/PPT 30					40(O)		100	3
S3	CB2005	Basics of IoT	20	30	10						40(W)		100	3
S4	CB2006	Object Oriented Programming		30	10							40 + 20	100	2
MDM	CBM00	Internet of Things				HA 30		35	35				100	3
HSS	HS2002	From Campus to Corporate - 1					50				50 (O)		100	2
RAD	HS2001	Reasoning and Aptitude Development - 3								100			100	1
DT	CB2001	Design Thinking – 1								100			100	1
EDI	CB2002	Engineering Design and Innovation - 1					30			70			100	2
Total			**70**	**90**	**30**	**60**	**80**	**35**	**35**	**270**	**170**	**60**	**900**	**21**

FF No.: 653

Structure

Branch: CSE-IoTCSBT Year: Second Year (SY) Academic Year 2025-26

Subject No.	Subject Code	Subject Name	Teaching Scheme(Hrs/Week))			Total	Credits
			Theory	Lab	Tut		
S1	CB2009	Data Structures and Analysis of Algorithms	2	2	0	4	3
S2	CB2010	Machine Learning	2	2	0	4	3
S3	CB2011	Database Management Systems	2	2	0	4	3
S4	CB2012	Computer Networks	2	0	1	3	3
MDM	MM0502	Software Project Management	2	0	1	3	3
HSS	HS2003	From Campus To Corporate - 2	2	0	0	2	2
RAD	HS2004	Reasoning and Aptitude Development - 4	1	0	0	1	1
DT	CB2013	Design Thinking – 2	0	0	1	1	1
ISMA	IR2201	OR International Design Thinking Module	-	-	-	-	1
EDI	CB2014	Engineering Design and Innovation – 2	0	4	0	4	2
IEDP	IR2202	OR International Engineering Design Program	-	-	-	-	2
		Total	**13**	**10**	**3**	**26**	**21**

Examination scheme

Subject No.	Subject Code	Subject Name	Examination scheme										Total	Credits
			CVV	CP	LAB CA	GD/ PPT/ HA	MSE (W/O)	T1 (W/O)	T2 (W/O)	ESE	ESE (W/O)	PRACT /CVV		
S1	CB2009	Data Structures and Analysis of Algorithms		30	10							40+20	100	3
S2	CB2010	Machine Learning	20	30	10						40(W)		100	3
S3	CB2011	Database Management Systems	20	30	10						40(W)		100	3
S4	CB2012	Computer Networks	30			GD/P PT 30					40(O)		100	3
MDM	MM050	Software Project Management				HA 30		35	35				100	3
HSS	HS2003	From Campus To Corporate - 2					50				50(O)		100	2
RAD	HS2004	Reasoning and Aptitude Development - 4								100			100	1
DT	CB2013	Design Thinking – 2								100			100	1
EDI	CB2014	Engineering Design and Innovation - 2					30			70			100	2
Total			**70**	**90**	**30**	**60**	**80**	**35**	**35**	**270**	**170**	**60**	**900**	**21**

FF No. : 654

CB2003: Fundamentals of Data Structures

Credits: 4 **Teaching Scheme**: Theory: 3 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

1. Knowledge of programming fundamentals using C or Python
2. Understanding of control structures, functions, and arrays
3. Basic concepts of memory management and file handling

Course Objectives:

1. To introduce students to linear and nonlinear data structures and their applications
2. To develop problem-solving skills by applying data structures to real-life scenarios
3. To understand and analyze the complexity of algorithms
4. To implement and apply various sorting and searching techniques
5. To develop efficient programs using stacks, queues, linked lists, trees, and graphs
6. To expose students to memory management, recursion, and performance trade-offs

Relevance of this course:

Data structure is the backbone of efficient software development, enabling programmers to store, organize, and process data effectively. By understanding and applying concepts such as arrays, linked lists, stacks, queues, trees, and graphs, students learn to design algorithms that solve real-world problems with optimal performance. This course builds foundational problem-solving and analytical skills essential for fields like systems programming, application development, artificial intelligence, and cyber security. Mastery of data structures equips students to write robust, maintainable, and scalable software, which is a critical competency for every computer engineer.

SECTION I

Unit I: Introduction to Data Structures and Complexity (7 Hours)

Abstract Data Types, need of data structures, classification of data structures, complexity analysis of algorithms using Big-O, Big-Ω and Big-Θ notations, time-space trade-off, recursion basics and applications

Unit II: Arrays and Searching & Sorting Techniques (7 Hours)

1D and 2D arrays, sparse matrix representation, linear search, binary search, bubble sort, selection sort, insertion sort, merge sort, quick sort, time complexity of all techniques

Unit III: Linked Lists (7 Hours)

Singly linked list, doubly linked list, circular linked list, operations (insert, delete, traverse, search), applications such as polynomial operations with different data structures, memory management using dynamic allocation

SECTION II

Unit IV: Stacks and Queues (7 Hours)

Stack operations and applications like expression evaluation, infix to postfix conversion, recursion using stack; queue operations, circular queue, priority queue, deque, applications of queues in real-world problems

Unit V: Trees (7 Hours)

Tree terminologies, binary trees, binary search trees (BST), tree traversals (inorder, preorder, postorder), expression trees, Huffman coding, applications in decision making and file systems

Unit VI: Graphs (7 Hours)

Graph representation using adjacency list and matrix, BFS, DFS, applications of graphs in networking and social media like transitive closure (Warshall's algorithm) and topological sorting

List of Practical:

1. Implement recursive functions and analyse their complexity. (Write a recursive function to generate all permutations of a string (e.g., password generator / anagram tool) OR Implement recursive solution to solve Tower of Hanoi and display moves + analyze time complexity)
2. Accept conventional matrix and convert it into sparse matrix using structure and Perform addition of two sparse matrices. Implement simple and fast transpose algorithms on sparse matrix.
3. Implement linear and binary search techniques (Search for a student's roll number in an unsorted list (linear) and sorted list (binary). Compare time taken OR Implement a phone contact search: use binary search to find contact names starting with given prefix)
4. Implement and compare sorting algorithms (bubble, insertion, selection) (Sort the set of strings in ascending order and descending order by using all 3 sorts (Display pass by pass output). Search a particular string using binary search with and without recursion. OR Sort a list of movies by release year and compare the performance of each algorithm)
5. Implement and compare sorting algorithms (merge, quick) (Sort large datasets (e.g., city temperatures over a month) and plot running time of both algorithms OR Sort names alphabetically using merge and quick sort, and count the number of comparisons)
6. Implement singly, doubly, and circular linked list (a) Create a playlist manager: add/delete/skip songs in a circular linked list (b) Maintain browsing history using doubly linked list (forward, backward navigation).
7. Implement polynomial addition using linked list (Store and add two polynomials representing real engineering measurements (e.g., signal processing) OR Develop a mini symbolic calculator to add and display polynomials entered by user)
8. Implement stack operations and applications (expression evaluation) (Evaluate postfix expressions from a calculator application OR Check if a given HTML/XML tag sequence is balanced (valid nesting))
9. Implement queue, circular queue, and dequeue (a)Simulate a print queue where print jobs arrive and are processed in FIFO order b) Simulate ticket booking counters using circular queue and dequeue (add, cancel, re-add)

10. Construct an expression tree from postfix/prefix expression and perform recursive inorder, preorder and post order traversals. For expression tree, perform non-recursive inorder, preorder and post order traversals.
11. Create a binary search tree (BST) of and perform following operations: i) Insert ii) Display inorder iii) Search a node iv) Find height of the tree v) level wise display iv) Delete v) Mirror
12. Create a graph using adjacency list representation. Perform graph traversal using BFS and DFS. OR Advanced assignment: Design a Maze Explorer where players are trapped in a maze and must use graph traversal to escape or collect treasures.
13. Mini Project (Trees / Graphs)

Text Books:

1. E. Horowitz, S. Sahni, D. Mehta; *Fundamentals of Data Structures in C*; 2nd Edition; Universities Press; 2008

2. Y. Langsam, M. Augenstein, A. Tannenbaum; *Data Structures Using C and C++*; 2nd Edition; Pearson Education; 2006

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein; *Introduction to Algorithms*; 3rd Edition; MIT Press; https://mitpress.mit.edu/9780262033848/

2. Brad Miller, David Ranum; *Problem Solving with Algorithms and Data Structures using Python*; Open Book Project; Accessed: May 2025; https://runestone.academy/ns/books/published/pythonds/index.html

e Learning Resources:

1. Data Structures and Algorithms Design By Prof. Nitin Saxena ; IIT Kanpur; https://onlinecourses.nptel.ac.in/noc25_cs81/preview

2. UC San Diego, National Research University Higher School of Economics; *Data Structures*; Coursera; https://www.coursera.org/learn/data-structures

Course Outcomes:

The student will be able to –

CO1: Analyse and calculate time and space complexity of given algorithms using standard notations, and validate performance through empirical testing.

CO2: Design and implement solutions to computational problems using appropriate searching and sorting algorithms, and evaluate their efficiency.

CO3: Develop programs using singly, doubly, and circular linked **lists** to manage and manipulate dynamic data effectively.

CO4: Apply stack and queue operations in real-world applications such as expression evaluation and scheduling problems.

CO5: Implement and apply tree data structures (binary trees, BST, AVL) for hierarchical data representation, traversal, and manipulation.

CO6: Implement graph data structures and apply graph traversal algorithms (BFS, DFS) to solve problems such as shortest path and connectivity analysis.

CO-PO Mapping Matrix: Fundamentals of Data Structures

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	_	_	_	_	_	2	_	3
CO2	3	3	3	2	2	_	_	_	_	2	_	2
CO3	2	2	3	2	2	_	_	_	_	2	_	2
CO4	2	3	3	2	2	2	_	_	_	2	_	2
CO5	3	3	3	2	2	2	_	_	2	2	_	2
CO6	3	3	3	2	2	3	_	_	2	2	_	3

Justification for Mapping for all COs

1. **CO1: Analyse and calculate time and space complexity of given algorithms using standard notations, and validate performance through empirical testing.**

    - **PO1**: Strongly mapped – Requires application of core computing knowledge to analyze algorithmic complexity.
    - **PO2**: Strongly mapped – Involves problem analysis and solution evaluation based on complexity.
    - **PO3**: Partially mapped – Algorithmic analysis guides effective design.
    - **PO4**: Partially mapped – Empirical testing aligns with modern analysis practices.
    - **PO10**: Mapped – Students document and communicate performance findings.
    - **PO12**: Strongly mapped – Students use modern tools for testing and validation.

2. **CO2: Design and implement solutions to computational problems using appropriate searching and sorting algorithms, and evaluate their efficiency.**

    - **PO1**: Strongly mapped – Students apply core knowledge to implement and understand algorithms.
    - **PO2**: Strongly mapped – Algorithm choice is based on problem characteristics and efficiency.
    - **PO3**: Strongly mapped – Involves designing functional and efficient software solutions.
    - **PO4**: Partially mapped – Comparative efficiency evaluation requires use of performance metrics.
    - **PO5**: Partially mapped – Involves practical implementation using modern platforms/tools.
    - **PO10**: Mapped – Includes documentation and clear communication of results.
    - **PO12**: Mapped – Encourages use of up-to-date development and testing tools.

3. **CO3: Develop programs using singly, doubly, and circular linked lists to manage and manipulate dynamic data effectively.**

    - **PO1**: Mapped – Involves understanding memory management and pointer operations.
    - **PO2**: Mapped – Students solve dynamic data problems using linked structures.
    - **PO3**: Strongly mapped – Involves design and coding of linked list-based systems.
    - **PO4**: Mapped – Includes debugging and verification of correctness.
    - **PO5**: Mapped – Uses software tools for development and visualization.
    - **PO10**: Mapped – Includes code documentation and explanation.
    - **PO12**: Mapped – Uses current development environments for implementation.

4. **CO4: Apply stack and queue operations in real-world applications such as expression evaluation and scheduling problems.**

    - **PO1**: Mapped – Students understand and apply abstract data types effectively.
    - **PO2**: Strongly mapped – Students select and apply ADTs to solve specific problems.
    - **PO3**: Strongly mapped – Develop and apply stack/queue solutions in real-world contexts.
    - **PO4**: Mapped – Evaluate performance of stack and queue operations under constraints.
    - **PO5**: Mapped – Practical implementation using programming environments.
    - **PO6**: Partially mapped – Real-world use cases show societal relevance (e.g., job queues).
    - **PO10**: Mapped – Includes reporting and communicating implementation logic.
    - **PO12**: Mapped – Tools and environments used for development and debugging.

5. **CO5: Implement and apply tree data structures (binary trees, BST, AVL) for hierarchical data representation, traversal, and manipulation.**

- **PO1**: Strongly mapped – Involves understanding recursive and hierarchical logic structures.
- **PO2**: Strongly mapped – Trees are applied to solve structured data problems (e.g., searching).
- **PO3**: Strongly mapped – Designing and implementing tree-based systems.
- **PO4**: Mapped – Evaluation of different tree algorithms (e.g., balancing) based on efficiency.
- **PO5**: Mapped – Uses IDEs and debuggers for implementing and testing trees.
- **PO6**: Partially mapped – Applications like file systems and parsers demonstrate societal relevance.
- **PO9**: Partially mapped – Tree-based projects may be team-based.
- **PO10**: Mapped – Requires written documentation and visual explanation.
- **PO12**: Mapped – Use of modern visual tools (tree visualization, debuggers).

6. **CO6: Implement graph data structures and apply graph traversal algorithms (BFS, DFS) to solve problems such as shortest path and connectivity analysis.**

- **PO1**: Strongly mapped – Involves mathematical and logical implementation of graphs.
- **PO2**: Strongly mapped – Students apply graphs to solve structured and networked problems.
- **PO3**: Strongly mapped – Graph structures are designed and implemented in software.
- **PO4**: Mapped – Empirical performance of graph traversal techniques is evaluated.
- **PO5**: Mapped – Uses current tools and graph libraries for implementation.
- **PO6**: Mapped – Real-world application in routing, maps, and network connectivity.
- **PO9**: Partially mapped – Graph-based projects may be team-based.
- **PO10**: Mapped – Results are communicated through code explanation and visual representation.
- **PO12**: Strongly mapped – Use of advanced graph tools, libraries, and simulators.

FF No. : 654

CB2004: Data Communication & Networking

Credits: 3 **Teaching Scheme:** Theory: 2 Hours/Week, Tut: 1 Hours/Week

Prerequisites:

1. Working of computer
2. Number System

Course Objectives:

1. To understand the functioning of data communication and computer network.
2. To provide an in-depth understanding of the OSI reference model.
3. To provide an in-depth understanding of the TCP/IP reference model.
4. To develop practical skills in designing and implementing different types of networks using IP addressing

Course Relevance:

This course equips students with hands-on skills in computer communication and Network system. It emphasizes real-time tool usage and applications of computer network.

Unit I: Data communication and its components (7 Hours)

Process of data communication and its components: Transmitter, Receiver, Medium, Message. Protocols, Standards, Standard organizations. Bandwidth, Data Transmission Rate, Baud Rate and Bits per second.
Modes of Communication (Simplex, Half duplex, Full Duplex). Signal & its properties.
Types of Errors: Single Bit Error and Burst Error, Redundancy
Error Detection: Longitudinal Redundancy Check (LRC), Vertical Redundancy Check (VRC), Cyclic Redundancy Check (CRC)
IEEE standards: 802.1, 802.2, 802.3, 802.4, 802.5 & Wireless LANs: 802.11 Architecture, MAC Sublayer

Unit II: OSI Reference Model (7 Hours)

OSI Reference Model: Layered Architecture, Peer-to-Peer Processes- Interfaces between Layer, Protocols, Organization of the Layers, Encapsulation.
Layers of the OSI Reference Model (Functions and features of each Layer) - Physical Layer, Data-Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, Application Layer.

Unit III: IP addressing (7 Hours)

Addressing mechanism in the Internet IP Addressing - IP Address classes, IP addressing, Subnetting, supernetting, Masking. IPv4 addressing.
IPv6 addressing – representation, address space allocation, Auto configuration.

Unit IV: TCP/IP Model (7 Hours)

TCP/IP Model: Layered Architecture, Data Link Layer: Nodes and links, services, categories of links, sub layers, Link layer addressing: three types of addresses, address resolution protocol (ARP).

Transport layer protocol: transport layer services, connectionless and connection-oriented protocol. TCP & UDP

List of Practical:

1. Network Topologies: Introduction, Definition, Selection, Criteria, Types of Topology- i) Bus ii) Ring iii) Star iv) Mesh v) Tree vi) Hybrid
2. Network Connecting Devices: Hub, Switch, Router, Repeater, Bridge, Gateway, Modem, Wireless infrastructure Components
3. Create desired standard network cable including cross cable and test by using cable Tester.
4. Connect Computers using given topology with wired media.
5. Connect Computers using wireless media.
6. Configure Static and Dynamic IP addresses
7. Configure DHCP server.
8. Run TCP/IP Utilities and Network commands: ipconfig, ping, tracert, netstat, pathping, route
9. Install Wireshark and configure as packet sniffer

Textbooks & References:

1. Data communications and networking -- Forouzan Behrouz A. -- Tata McGraw Hill, New Delhi,
2. Computer Networks-Tanenbaum Andrew S.—Publication--PHI Learning Pvt Ltd, Delhi
3. Data Communication and Networks -- Godbole Achyut -- Tata McGraw Hill, New Delhi

e Learning Resources:

CCNA: Introduction to Networks: https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US

Course Outcomes:

The student will be able to –

CO1: Explain and apply the fundamentals of data communication and networking concepts such as transmission media, protocols, and network devices.

CO2: Analyze and differentiate the functions of each layer in the OSI reference model using real-world examples and scenarios.

CO3: Analyze the TCP/IP network communication model and evaluate its layered structure through case studies and configuration exercises.

CO4: Design and implement a small computer network by assigning and configuring IP addresses, subnet masks, and basic routing.

CO-PO Mapping Matrix:

CO\ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**
CO2	**3**	**2**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO3	**3**	**2**	**-**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO4	**3**	**-**	**3**	**-**	**3**	**-**	**-**	**-**	**2**	**-**	**2**	**-**

Scale:
3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping

1. **CO1: Explain and apply the fundamentals of data communication and networking concepts such as transmission media, protocols, and network devices.**

   - **PO1 – Engineering knowledge**: Understanding data communication principles requires foundational engineering knowledge in electronics, signals, and computer systems.
   - **PO2 – Problem analysis**: Students analyze and choose appropriate transmission media or devices based on requirements, demonstrating problem-solving ability.
   - **PO3 – Life-long learning**: Networking technologies evolve rapidly; understanding fundamentals encourages continual learning and adaptation.

2. **CO2: Analyze and differentiate the functions of each layer in the OSI reference model using real-world examples and scenarios.**

- **PO1 – Engineering knowledge**: Knowledge of OSI layers stems from established engineering principles in networking and communication systems.
- **PO2 – Problem analysis**: Differentiating functions across layers involves analyzing real-world communication problems.
- **PO3 – Conduct investigations of complex problems**: Analyzing issues across OSI layers requires investigation and critical thinking, especially for troubleshooting layered interactions.

3. **CO3: Analyze the TCP/IP network communication model and evaluate its layered structure through case studies and configuration exercises.**

- **PO1 – Engineering knowledge**: Mastery of TCP/IP requires application of core engineering knowledge in computer systems and protocols.
- **PO2 – Problem analysis**: Students analyze packet flow, protocol behavior, and troubleshoot communication errors in layered TCP/IP structure.
- **PO3 – Conduct investigations of complex problems**: Case studies demand detailed investigations into protocol operations and network performance.
- **PO4 – Modern tool usage**: Configuration exercises involve use of tools like Wireshark, Packet Tracer, or real network setups.

4. **CO4: Design and implement a small computer network by assigning and configuring IP addresses, subnet masks, and basic routing.**

- **PO1 – Engineering knowledge**: Designing a network involves applying mathematical and technical concepts such as binary arithmetic and IP addressing.
- **PO2 – Design/development of solutions**: Students design and build small-scale network systems as practical solutions to communication needs.
- **PO3 – Modern tool usage**: Implementation uses modern tools for configuration and testing such as routers, switches, and simulators.
- **PO4 – Individual and teamwork**: Network setup projects are often collaborative, involving team planning and execution.
- **PO5 – Project management and finance**: Planning IP allocation, choosing devices, and managing resources reflects basic project management skills.

CB2005: Basics of IoT

Credits: 3 **Teaching Scheme:** Theory: 2 Hours/Week, Practical (P): 2 hrs. /week

Prerequisites:

1. Digital System Design
2. Electronics Workshop
3. Fundamentals of IoT

Course Objectives:

1. To study fundamental concepts of IoT
2. To Learn different protocols used for IoT design
3. To be familiar with data handling and analytics tools in IoT.
4. To give an overview of 8 bit architecture of Microcontroller.

Course Relevance:

This course equips students with hands-on skills in Internet of things concepts and Network applications. It emphasizes real-time tool usage and applications of IoT

Unit I: Introduction to IoT (7 Hours)

Introduction, Definitions & Characteristics of IoT, History of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M.

Unit II: Introduction of Microprocessor & Microcontroller (7 Hours)

Basics of Microprocessor, Types & evolution, Block diagram & functioning, Evolution of microcontrollers, Microcontroller selection criteria for particular application, MCS-51 architecture, family devices & its derivatives. Pin configuration, Port architecture, memory organization, external memory interfacing.

Unit III: IP based Protocols for IOT (7 Hours)

IPv6, 6LowPAN, RPL, REST, AMPQ, CoAP, MQTT.
Authorization and Access Control in IOT

Unit IV: IoT Security and Privacy (7 Hours)

Challenges and threats to IoT security, Encryption and authentication techniques, Privacy concerns and regulations, Best practices for securing IoT devices

List of Practical:

1. Study& Survey of various development boards for IoT.
2. Study & Survey of various IoT platforms.

3. Interfacing sensors and actuators with Arduino Uno
4. Build a cloud-ready temperature sensor with the Arduino Uno and the any IoT Platform: This project shows the building of a temperature sensor.
5. Interfacing Sensors and actuators with Arduino Uno
6. IoT based Stepper Motor Control with Raspberry Pi.
7. IoT based Web Controlled Home Automation using Arduino Uno
8. A Simple IoT Project with the ESP8266 WiFi module
9. Implement a RFID Based IoT Project

Course Outcomes:

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

CO1: Explain and demonstrate the use of concepts, terminologies, and architecture of IoT systems through case studies and system design examples.

CO2: Compare and analyze the working principles of microprocessors and microcontrollers through practical experiments and simulations

CO3: Apply appropriate communication protocols for the design and implementation of IoT systems in real-world scenarios.

CO4: Identify and apply IoT security and privacy methods to secure IoT devices, data, and communication channels in given problem statements.

Textbooks & References:

1. Hakima Chaouchi, " The Internet of Things Connecting Objects to the Web" ISBN : 978-1- 84821-140-7, Wiley Publications
2. Olivier Hersent, David Boswarthick, and Omar Elloumi, "The Internet of Things: Key Applications and Protocols", WileyPublications
3. Internet of Things, Arsheep Bahga and Vijay Madisetti.
4. Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications

e Learning Resources:

1. Introduction To Internet of Things, NPTEL course By Prof. Sudip Misra, IIT Kharagpur, https://onlinecourses.nptel.ac.in/noc22_cs53/preview
2. IoT Devices, Infosys Springboard course, https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0142354096914759682649/overview

CO-PO Mapping Matrix:

CO\ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	–	–	–	–	–	_	-	_	2
CO2	3	–	–	3	3	–	–	-	–	–	-	–
CO3	3	–	3	–	3	–	–	–	-	–	-	–
CO4	3	3	3	–	–	2	-	2	–	–	-	-

Scale:
3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping

1. **CO1: Explain and demonstrate the use of concepts, terminologies, and architecture of IoT systems through case studies and system design examples.**

    - **PO1 – Engineering Knowledge**: This CO involves understanding the basic building blocks of IoT systems, which draws on foundational concepts in electronics, networking, and computing.
    - **PO2 – Problem Analysis**: Case studies require learners to analyze existing IoT systems, identify the underlying problems or requirements, and understand how solutions were built.
    - **PO3 – Design/Development of Solutions**: Demonstrating IoT system architectures through examples builds design-thinking skills aligned with real-world system development.
    - **PO4 – Life-long Learning**: As IoT technologies rapidly evolve, learning through case studies fosters a habit of continuous and independent learning.

2. **CO2: Compare and analyze the working principles of microprocessors and microcontrollers through practical experiments and simulations.**

- **PO1 – Engineering Knowledge**: Understanding and comparing microprocessors and microcontrollers is a core element of embedded systems knowledge.
- **PO2 – Conduct Investigations of Complex Problems**: Practical experiments and simulations help investigate how these devices behave under various conditions, contributing to analytical problem-solving.
- **PO3 – Modern Tool Usage**: This CO involves using simulators and development environments (e.g., Arduino, Keil, Proteus), reflecting the application of modern engineering tools.

3. **CO3: Apply appropriate communication protocols for the design and implementation of IoT systems in real-world scenarios.**

- **PO1 – Engineering Knowledge**: Selecting and applying protocols like MQTT, CoAP, or HTTP requires solid knowledge of communication principles and IoT requirements.
- **PO2 – Design/Development of Solutions**: The ability to choose suitable protocols impacts the overall design and effectiveness of the IoT solution.
- **PO3 – Modern Tool Usage**: Implementing real-world protocols using IoT hardware and platforms involves configuring and using modern tools and SDKs.

4. **CO4: Identify and apply IoT security and privacy methods to secure IoT devices, data, and communication channels in given problem statements.**

- **PO1 – Engineering Knowledge**: Applying encryption, authentication, and data protection methods relies on a strong foundation in computer networks and cybersecurity.
- **PO2 – Problem Analysis**: Analyzing threats, vulnerabilities, and security requirements in IoT problem statements requires critical problem-solving skills.
- **PO3 – Design/Development of Solutions**: Implementing appropriate security solutions demonstrates the ability to design systems that address identified threats.
- **PO4 – The Engineer and Society**: Securing IoT devices and data has a direct impact on user safety, societal welfare, and ethical data handling.
- **PO5 – Ethics**: Privacy-preserving designs align with ethical responsibilities in engineering, particularly when handling sensitive user data.

FF No. : 654

CB2006: Object Oriented Programming

Credits: 2 **Teaching Scheme:** Theory: 1 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

1. Basic Programming

Course Objectives:

1. To introduce the fundamentals of object-oriented programming and familiarize students with the basic syntax and features of C++.
2. To develop the ability to implement modular, maintainable, and reusable code using classes, objects, constructors, destructors, and operator overloading.
3. To explain and apply core OOP concepts such as inheritance, polymorphism, function overloading, and virtual functions for building hierarchical and dynamic programs.
4. To enable students to perform file operations (text and binary) and handle runtime errors using exception handling mechanisms in C++.
5. To introduce templates for generic programming and promote writing efficient and type-independent code.

Course Relevance:

This course equips CSE-IoTCSBT students with essential object-oriented programming skills using C++, which are crucial for developing efficient, modular, and reusable software for IoT systems. Concepts like classes, inheritance, and polymorphism help in designing hardware-abstracted and scalable code. File handling and exception management enable robust data processing and error control in real-time IoT applications. Templates promote reusable code for drivers and protocols. Overall, it builds a strong foundation for embedded programming and advanced IoT development.

Unit I: Introduction to OOP

Fundamentals of OOPS, Introduction to Programming and C++, How C++ differs from C,Variables, Data Types, and Operators , Control Structures, Loops and Iteration, Functions and Modular Programming, Basics of Console Input and Output Class, Dynamic Memory Allocation
Overview of OOPs Principles , Introduction to classes & objects , Creation & destruction of objects , Data Members , Member Functions , Access Specifier, this Pointer , Constructor &Destructor ,

Static class member ,Friend class and functions , Function Overloading, Operator Overloading Namespace.

Unit II: OOP Principles

Introduction to inheritance, Base and Derived class Constructors , Types of Inheritance, Down casting and up casting, Function overriding, Virtual functions, Polymorphism, Pure virtual functions, Virtual Base Class, C++ Class Hierarchy , File Stream ,Text File Handling , Binary File Handling , Error handling during file operations , Overloading << and >> operators, Introduction to Exception, Benefits of Exception handling, Try and catch block, Throw statement, Pre-defined exceptions in C++, Writing custom Exception class, Stack Unwinding, Function Templates , Class Templates, Standard Template Library (STL)

List of Practical:

1. Basics of C++
    - Write a C++ program to calculate the area of a rectangle given its length and width.
    - Develop a C++ program that converts temperature from Celsius to Fahrenheit using the formula: Fahrenheit = (Celsius * 9/5) + 32.
    - Create a program that takes a user's age as input and determines if they are eligible to vote or not.
    - Implement a C++ program that generates the Fibonacci sequence up to a given number 'n' using loops.
2. Functions and Modular Programming
    - Design a program that calculates the factorial of a given positive integer using a recursive function.
    - Develop a modular program that checks whether a given number is prime or not, utilizing a function for prime number testing.
3. Object-Oriented Programming (OOP)
    - Create a C++ class named Rectangle that has attributes for length and width. Implement methods to calculate the area and perimeter of the rectangle.
    - Design a program using OOP concepts to simulate a basic banking system. Implement classes for customers and accounts, allowing for deposits and withdrawals. Use Constructors.
    - Write a program to use static data members and member functions.
    - Use this pointer to resolve naming conflicts.
    - Create a program to overload arithmetic operators using friend functions.
    - Implement overloading of comparison operators (==, <, etc.).
4. Inheritance and Polymorphism

- Build a hierarchy of classes representing different shapes (e.g., Circle, Triangle, Rectangle) with a common base class. Implement a virtual function for calculating the area of each shape.
- Extend the banking system to include different types of accounts (Savings, Checking) that inherit from a common Account class. Implement polymorphic behavior for interest calculations.

5. Write a program using try, catch, and throw for arithmetic and input errors.
6. Write a function template for finding the maximum of two values.

Textbooks & References:

1. Behrouz A. Forouzan, Richard F. Gilberg, "COMPUTER SCIENCE – A Structred Programming approach using C", Indian Edition, Thomson, 3rd edition
2. BjarneStroustrup, ― The C++ Programming language‖, Third edition, Pearson Education. ISBN 9780201889543
3. Kernighan, Ritchie, "The C Programming Language", Prentice Hall of India
4. Robert Lafore, ―Object-Oriented Programming in C++, fourth edition, Sams Publishing, ISBN:0672323087 (ISBN 13: 9780672323089)
5. Herbert Schildt, ―C++ The complete reference‖, Eighth Edition, McGraw Hill Professional, 2011, ISBN:978-00-72226805
6. E. Balagurusamy-- Object-oriented programming with C++, fourth edition, Mc Hill Professional,2008, ISBN 978-0-07-066907-9

e Learning Resources:

1. Swayam Course -Programming in Modern C++, By Prof. Partha Pratim Das | IIT Kharagpur https://onlinecourses.nptel.ac.in/noc25_cs144/preview
2. Infosys Springboard course- Programming Using C++ ,By Malarvizhi Rathinam, Balasundari Vaidyanathan https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01297200240671948837_shared/overview

Course Outcomes:

Students will be able to:

CO1: Design and implement modular C++ programs applying object-oriented programming fundamentals, including classes, objects, constructors, destructors, and operator overloading.
CO2: Design and develop C++ applications incorporating inheritance, polymorphism, file handling, exception handling, and templates to create robust and reusable software solutions.

CO-PO Mapping Matrix: Object Oriented Programming

CO\ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	**3**	**2**	**3**	**-**	**2**	**-**	**-**	**-**	**1**	**-**	**-**	**2**
CO2	**3**	**2**	**3**	**-**	**2**	**1**	**-**	**-**	**-**	**-**	**1**	**2**

Scale:

3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping

1. **CO1: Design and implement modular C++ programs applying object-oriented programming fundamentals, including classes, objects, constructors, destructors, and operator overloading.**

- **PO1 (Engineering Knowledge):** Requires knowledge of object-oriented principles like classes, objects, and constructors to implement modular and scalable software.
- **PO2 (Problem Analysis):** Students analyze a problem, decompose it into classes and objects, and identify design strategies.
- **PO3 (Design/Development of Solutions):** Core activity here is designing software solutions using OOP principles.
- **PO5 (Modern Tool Usage):** May involve the use of IDEs/debuggers and version control tools when writing and testing C++ programs.

- **PO9 (Individual and Team Work):** Modular program design supports collaboration; students may work in teams for implementation and testing.
- **PO12 (Lifelong Learning):** Learning object-oriented paradigms forms the foundation for advanced software development skills, encouraging continued learning.

2. **CO2: Design and develop C++ applications incorporating inheritance, polymorphism, file handling, exception handling, and templates to create robust and reusable software solutions.**

- **PO1 (Engineering Knowledge):** Applies foundational programming knowledge and advanced C++ features.
- **PO2 (Problem Analysis):** Analyzing the requirements to choose appropriate features like inheritance or templates for solving problems.
- **PO3 (Design/Development of Solutions):** Involves developing efficient, reusable, and robust solutions using polymorphism and exception handling.
- **PO5 (Modern Tool Usage):** Use of development environments, debugging tools, and possibly static analysis tools to enhance robustness.
- **PO6 (Engineer and Society):** Robust software often considers safe handling of exceptions, which aligns with responsible engineering practices.
- **PO11 (Project Management and Finance):** Understanding reusability and robustness through templates and inheritance improves time and resource management in software projects.
- **PO12 (Lifelong Learning):** Encourages mastery of advanced C++ concepts, preparing learners for evolving programming paradigms.

FF No. : 654

CBM001: Internet of Things

Credits: 3 **Teaching Scheme:** Theory: 2 Hours/Week, Tut: 1 Hours/Week

Prerequisites

1. Digital System Design
2. Electronics Workshop
3. Fundamentals of IoT

Course Objectives

1. To study fundamental concepts of IoT
2. To Learn different protocols used for IoT design
3. To be familiar with data handling and analytics tools in IoT.
4. To give an overview of 8 bit architecture of Microcontroller.

Course Relevance

This course equips students with hands-on skills in Internet of things concepts and Network applications. It emphasizes real-time tool usage and applications of IoT

Unit I: Introduction to IoT (7 Hours)

Introduction, Definitions & Characteristics of IoT, History of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M.

Unit II: Types of IoT Sensors & Actuators (7 Hours)

Types of IoT Sensors:- Temperature sensors, Humidity sensors, Motion sensors, Gas sensors, Smoke sensors, Pressure sensors, Image sensors, Accelerometer sensors, IR sensors, Proximity sensors, Torque sensors.

Basic actuators:- Servo motors, Stepper Motor, DC motors, Linear Actuator, Relay, Solenoid.

Unit III: Applications (7 Hours)

Use of Smart Sensors in IOT enabled devices and in Industry 4.0, Application area of Smart Sensors. Need of Smart Sensors in IOT, ROBOTICS and Modern industrial applications

Unit IV: IoT Security and Privacy (7 Hours)

Challenges and threats to IoT security, Encryption and authentication techniques, Privacy concerns and regulations, Best practices for securing IoT devices

List of Practical:

1. Study& Survey of various development boards for IoT.
2. Study & Survey of various IoT platforms.
3. Interfacing sensors and actuators with Arduino Uno
4. Build a cloud-ready temperature sensor with the Arduino Uno and the any IoT Platform: This project shows the building of a temperature sensor.
5. Interfacing Sensors and actuators with Arduino Uno
6. IoT based Stepper Motor Control with Raspberry Pi.
7. IoT based Web Controlled Home Automation using Arduino Uno
8. A Simple IoT Project with the ESP8266 WiFi module
9. Implement a RFID Based IoT Project

Textbooks & References:

1. Hakima Chaouchi, " The Internet of Things Connecting Objects to the Web" ISBN : 978-1- 84821-140-7, Wiley Publications
2. Olivier Hersent, David Boswarthick, and Omar Elloumi, "The Internet of Things: Key Applications and Protocols", WileyPublications
3. Internet of Things, Arsheep Bahga and Vijay Madisetti.
4. Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications

e Learning Resources:

1. Introduction To Internet of Things, NPTEL course By Prof. Sudip Misra, IIT Kharagpur, https://onlinecourses.nptel.ac.in/noc22_cs53/preview
2. IoT Devices, Infosys Springboard course, https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0142354096914759682649/overview

Course Outcomes:

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

CO1: Explain and demonstrate the use of concepts, terminologies, and architecture of IoT systems through case studies and system design examples.

CO2: Classify various types of sensors, actuators and describe their operational characteristics when used in real world IoT systems.

CO3: Analyze the role and necessity of smart sensors in IoT-based applications across domains like Industry 4.0, robotics, and modern industrial systems.

CO4: Identify and apply IoT security and privacy methods to secure IoT devices, data, and communication channels in given problem statements.

CO-PO Mapping Matrix:

CO\ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3									2
CO2	3	2	3		3							2
CO3	2	3	3				2					2
CO4	3	3	3			2		2				

Scale:
3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping

1. **CO1: Explain and demonstrate the use of concepts, terminologies, and architecture of IoT systems through case studies and system design examples.**

   - **PO1 – Engineering Knowledge**: This CO involves understanding the basic building blocks of IoT systems, which draws on foundational concepts in electronics, networking, and computing.
   - **PO2 – Problem Analysis**: Case studies require learners to analyze existing IoT systems, identify the underlying problems or requirements, and understand how solutions were built.
   - **PO3 – Design/Development of Solutions**: Demonstrating IoT system architectures through examples builds design-thinking skills aligned with real-world system development.
   - **PO12 – Life-long Learning**: As IoT technologies rapidly evolve, learning through case studies fosters a habit of continuous and independent learning.

2. **CO2: Classify various types of sensors, actuators and describe their operational characteristics when used in real-world IoT systems.**

   - **PO1**: Applies engineering fundamentals to classify sensors and actuators.
   - **PO2**: Engages in problem analysis by evaluating sensor/actuator suitability for real-world applications.

- **PO4**: Requires interpretation of data sheets and performance metrics (research-based knowledge).
- **PO5**: Utilizes modern tools for simulating or evaluating sensor behavior in IoT.
- **PO12**: Encourages continuous learning about evolving sensor technologies and their applications.

3. **CO3: Analyze the role and necessity of smart sensors in IoT-based applications across domains like Industry 4.0, robotics, and modern industrial systems.**

- **PO1**: Understanding domain-specific IoT applications requires fundamental engineering knowledge.
- **PO2**: Analysis of application-specific smart sensors in various industries.
- **PO3**: Relates to design and development of IoT-based systems using smart sensors.
- **PO7**: Understanding the societal and environmental impact of deploying IoT in domains like Industry 4.0.
- **PO12**: Promotes awareness of technological trends and encourages lifelong learning.

4. **CO4: Identify and apply IoT security and privacy methods to secure IoT devices, data, and communication channels in given problem statements.**

- **PO1 – Engineering Knowledge**: Applying encryption, authentication, and data protection methods relies on a strong foundation in computer networks and cybersecurity.
- **PO2 – Problem Analysis**: Analyzing threats, vulnerabilities, and security requirements in IoT problem statements requires critical problem-solving skills.
- **PO3 – Design/Development of Solutions**: Implementing appropriate security solutions demonstrates the ability to design systems that address identified threats.
- **PO6 – The Engineer and Society**: Securing IoT devices and data has a direct impact on user safety, societal welfare, and ethical data handling.
- **PO8 – Ethics**: Privacy-preserving designs align with ethical responsibilities in engineering, particularly when handling sensitive user data.

HS2002: From Campus To Corporate – 1

Credits:.2 **Teaching Scheme: Theory: 2 hours/Week**

Introduction to the Corporate World Understanding organizational structure and hierarchy, Work culture differences: campus vs. corporate, Employer expectations from fresh graduates, Time management and ownership in corporate settings

Professional Communication Skills: Verbal and non-verbal communication, Email and business writing etiquette, Presentation skills and use of visual aids, Listening skills and telephone etiquette,

Soft Skills and Interpersonal Effectiveness: Body language, grooming, and first impressions, Conflict resolution and negotiation skills, Team dynamics and collaboration, Assertiveness vs. aggressiveness

Resume Building and Job Preparation : Building an effective resume and cover letter, Identifying strengths and achievements, Preparing for technical and HR interviews, Handling rejections and feedback

Group Discussions and Personal Interviews : Group discussion formats and evaluation criteria, Strategies for initiating, contributing, and summarizing, Mock interviews with feedback, STAR technique for answering behavioral questions,

Corporate Etiquette and Workplace Ethics: Meeting and greeting protocol, Dining and social etiquette, Work ethics, punctuality, confidentiality, Respect for diversity and inclusion in the workplace

Adaptability and Emotional Intelligence: Handling pressure, deadlines, and ambiguity, Self-awareness and emotional regulation, Empathy and workplace relationships, Managing feedback and continuous learning,

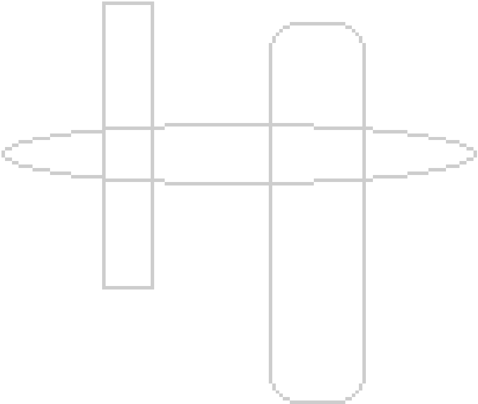
Introduction to Project Management Basics: Understanding tasks, milestones, deadlines, Collaboration using tools like Trello, Slack, Teams, Basics of Agile/Scrum concepts, Reporting and escalation protocol

Faculty are supposed to do conduct following in the class

- Resume and LinkedIn profile workshops
- Mock interviews and GD sessions
- Role plays: workplace scenarios, conflict handling
- Business email writing exercises
- Presentation and elevator pitch sessions

Books:

1. Dale Carnegie, How to Win Friends and Influence People
2. Stephen R. Covey, 7 Habits of Highly Effective People
3. Shital Kakkar Mehra, Business Etiquette: A Guide for the Indian Professional
4. Peggy Klaus, The Hard Truth About Soft Skills



FF No. : 654

HS2001: Reasoning and Aptitude Development - 3

Credits: 1 **Teaching Scheme:** 1 Hour/Week

Unit 1: English Language

Familiarity with English Language, Ability to understand written text, spoken word and effective communication through written documents; Coverage of vocabulary to cope up with general and specific terminology, syntax and sentence structure, prevention of incorrect use leading to distortion in communication; synonyms, antonyms and contextual vocabulary, Grammar – Error identification, sentence improvement and construction, Reading Comprehension

Unit 2: Logical Ability

Objective interpretation of things, ability to perceive and interpret trends to make generalizations; ability to analyze assumptions behind an argument or statement; Deductive reasoning: Assessment of ability to synthesize information and derive conclusions - Coding deduction logic, Data Sufficiency, Directional Sense, Logical word sequence, Objective reasoning, Selection and decision tables, puzzles; Inductive reasoning: Assessment of ability to learn by example, imitation or by trial – Analogy pattern recognition, Classification pattern recognition, Coding pattern recognition, Number series pattern recognition; Abductive reasoning: Critical thinking ability of seeing through logical weak links or loopholes in an argument or a group of statements; Critical reasoning: assessment of ability to think through and analyze logical arguments, assessment of ability to use logical constructs to offer reasoning in unfamiliar situations; Information Gathering and synthesis: Ability of locating information, information ordering, rule based selection and data interpretation, order and classify data, interpret graphs, charts, tables and make rule based deductions. Application of these approaches for using visual, numerical and textual data from single or multiple sources

Unit 3: Quantitative Ability

Basic numbers – decimals and fractions, factorization, divisibility: HCF, LCM, Odd, even, prime and rational numbers. Application of algebra to real world, direct and inverse proportion, common applications – Speed-time -distance, Profit-loss, percentage, age relations, mixtures, other miscellaneous quantitative combination, exponentials and logarithms, permutations and combinations, probability. Spatial reasoning: Inductive – Missing portions, Sequence and series; Deductive analysis.

Reference Books –

1. "English Grammar in Use" by Raymond Murphy, Cambridge University Press.
2. "Word Power Made Easy" by Norman Lewis, Goyal Publishers & Distributors.
3. "Objective General English" by S.P. Bakshi, Arihant Publications.
4. "English for Competitive Examinations" by K. Sinha, S. Chand Publishing.
5. "Essential English Grammar" by Philip Gucker, Wiley.
6. "English Idioms and Phrasal Verbs" by M.A. Yadav, Vikas Publishing House.
7. "The Oxford English Grammar" by Sidney Greenbaum, Oxford University Press.
8. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121903409.

9. "Logical Reasoning and Data Interpretation for the CAT" by Nishit K. Sinha, Pearson India, ISBN: 978-8131709117.
10. "Logical Reasoning and Data Interpretation for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
11. "A New Approach to Reasoning Verbal and Non-Verbal" by B.S. Sijwali & Indu Sijwali, Arihant Publications, ISBN: 978-9311124692.
12. "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121900637.
13. "How to Prepare for Quantitative Aptitude for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
14. "The Pearson Guide to Quantitative Aptitude for Competitive Examination" by Pearson, Pearson India, ISBN: 978-8131709117.
15. "Quantitative Aptitude for Competitive Examinations" by Abhijit Guha, Tata McGraw Hill Education, ISBN: 978-0070666653.
16. "Data Interpretation & Data Sufficiency" by R.S. Aggarwal, S. Chand Publishing ISBN: 978-8121903515.
17. "Quantitative Aptitude for Competitive Examinations" by S. Chand, S. Chand Publishing, ISBN: 978-8121903423.

Course Outcomes:

Upon completion of the course, the student will be able to –

1. Improve the reading, writing and verbal skills, and enhance comprehension and articulation abilities
2. Develop logical reasoning abilities, enabling them to make sound decisions in problem-solving scenarios
3. Develop mathematical aptitude as well as data interpretation abilities and use them in test cases and real world problems
4. Learn to apply approaches for optimum time-management, prioritization maximizing the accuracy
5. Learn data interpretation, apply mathematical skills to draw accurate conclusions
6. Apply their knowledge of English, reasoning and quantitative skills for planning, critical thinking and real world problems

CO-PO Map

	Program Outcomes (PO)												**PSO**			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2					3		2	2				3
CO2	2	2	3	2	2		2		3		2	2	3		3	3
CO3	2	2	3	2	3		2		3		2	2	3		3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3	3	3
CO5	2	2	3	2	3	2			3		2	2	3		3	3
CO6	2	2	3	3	2				3		3	2	3		3	3
Average	**2.0**	**2.0**	**2.83**	**2.83**	**2.6**	**2.5**	**2.0**	**2.0**	**3.0**	**1.0**	**2.16**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**

FF No. : 654

CB2001: Design Thinking - 1

Credits: 1 **Teaching Scheme:** Tutorial: 1 Hour/Week

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:
To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?
Importance of Paper Publication and Patents
Structure of Paper
Journal Publication
Publication in conference
Literature Review
Research Paper Writing
Journal Ratings and Evaluation
How to rate a Journal?
Intellectual property (IP)
Research Ethics
Entrepreneurship

Section 2: Topics/Contents

Structure of The paper
Journal List (Top 50 Journals)
Selection of the journal
Use of various online journal selection tools
Plagiarism checking
Improving contents of the paper
Patent drafting
Patent search
Filing of patent
Writing answers to reviewer questions
Modification in manuscript
Checking of publication draft

Course Outcomes: **[**Publication of paper or patent]
The student will be able to
1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research

7. Understand the Entrepreneurship and Business Planning

CO-PO Map:

	Program Outcomes (PO)												**PSO**			
CO/PO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	3
CO2	1	1	1	1	1	--	--	--	--	--	--	1	2	1	1	3
CO3	2	2	3	3	2	2	1	2	2	3	--	1	2	2	3	3
CO4	3	3	3	3	3	2	1	2	2	3	1	1	-	-	2	3
CO5	1	1	1	1	1	--	--	--	--	--	--	1	-	-	1	2
CO6	2	2	2	2	2	2	1	3	2	3	--	1	2	2	2	3
C07	1	1	1	1	1	--	--	--	--	--	--	1	1	1	1	1
Average	**1.57**	**1.57**	**1.71**	**1.71**	**1.57**	**2.0**	**1.0**	**2.33**	**2.0**	**3.0**	**1.0**	**1.0**	1.66	1.66	1.71	2.5

FF No. : 654

CB2002: Engineering Design and Innovation - 1

Credits:2 **Teaching Scheme:** 4 Hours/Week

Course Prerequisites: Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project centric learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Engineering Design" are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:
Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.
- It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution.

How effective is …….? How strong is the evidence for ………? How clear is ……?

- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDD Sample Case Studies : -

With the adaptation of industry communication standards, Raspberry Pi and Sensors, following projects can be taken up:

1) Design a deployable product for soil moisture detection
2) Design a deployable product for temperature detection
3) Design a deployable product for pressure detection
3) Design a deployable product smoke detection
4) Design a deployable product for motion detection
5) Design a deployable product for collision detection
6) Design a deployable product for sound detection

...not limited to.....Faculty and students are free to include other areas which meet the society requirements at large.

Text Books: ***(As per IEEE format)***

1.A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017
2.Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.
3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: ***(As per IEEE format)***

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

MOOCs Links and additional reading material:
www.nptelvideos.in
https://worldwide.espacenet.com/

Course Outcomes:

On completion of the course, learner will be able to–

1. Identify the real-life problem from societal need point of view
2. Choose and compare alternative approaches to select most feasible one
3. Analyze and synthesize the identified problem from technological perspective
4. Design the reliable and scalable solution to meet challenges
5. Evaluate the solution based on the criteria specified
6. Inculcate long life learning attitude towards the societal problems

CO PO Map

	Program Outcomes (PO)												**PSO**			
CO/PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**
CO1	2	2	2	2					3		2	2				3
CO2	2	2	3	2	2		2		3		2	2	3		3	3
CO3	2	2	3	2	3		2		3		2	2	3		3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3	3	3
CO5	2	2	3	2	3	2			3		2	2	3		3	3
CO6	2	2	3	3	2				3		3	2	3		3	3
Average	**2.0**	**2.0**	**2.83**	**2.83**	**2.6**	**2.5**	**2.0**	**2.0**	**3.0**	**1.0**	**2.16**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

FF No. : 654

CB2003: Data Structures and Analysis of Algorithms

Credits: 3 **Teaching Scheme**: Theory: 2 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

1. Fundamentals of data structures
2. Object oriented programming

Course Objectives:

1. To study advanced data structures such as hash table, AVL tree
2. To know the basics of computational complexity analysis and various algorithm design strategies.
3. To provide students with solid foundations to deal with a wide variety of computational problems.
4. To analyze an algorithmic strategy and identify the computing requirements appropriate for its solutions.
5. To understand basic concepts of P, NP class problems and parallel algorithms

Relevance of this course:

Data Structures and Analysis of Algorithms is a core foundational course in computer science that equips students with the ability to organize data efficiently and design algorithms with optimal performance. The course develops analytical and problem-solving skills required to evaluate time and space complexity, enabling students to choose appropriate data structures and algorithms for real-world applications. Knowledge gained in this course is essential for advanced studies in software engineering, databases, artificial intelligence, machine learning, operating systems, networking, and competitive programming, and is highly valued by the software industry for developing scalable and efficient solutions**.**

SECTION I

Unit I: Advanced data structures (7 Hours)

Hash tables: Basic concepts, Hash function, Characteristics of good hash function, Different key-to-address transformations techniques, Synonyms or collisions, Collision resolution techniques-linear probing, Quadratic probing, Rehashing, Chaining without replacement and Chaining with replacement, Applications.
Heap: Heap definition, Heap properties, Types of heap, Heap data structure, heap sort implementation, priority queue using heap
Threaded binary tree, OBST, AVL trees, Red-Black trees, B-Tree, B+ tree, Trie.

Unit II: Greedy method (7 Hours)

Concept of Greedy method, Comparison with Divide-and-Conquer strategy (recurrence relations), Minimum spanning tree: Prim’s and Kruskal’s algorithms, shortest path using Dijkstra’s algorithm

SECTION II

UNIT III: Dynamic Programming and Backtracking (7 Hours)

Concept of dynamic programming: Travelling Salesman Problem, Floyd Warshall Algorithm, Memorization Strategy. 0/1 Knapsack Problem, longest common subsequence.
Backtracking: General method, Recursive backtracking algorithm, iterative backtracking method. N- queens problem, Sum of subsets, 0/1 Knapsack Problem, Graph Coloring.

Unit IV: Branch and Bound and Computational Complexities (7 Hours)

Branch and bound method, Control abstraction for Least Cost Search, Bounding, FIFO branch and bound, LC Branch and bound, 0/1 Knapsack problem -LC branch and bound and FIFO branch and bound solution.
The classes P, NP, NP Complete and NP-hard: Basic concepts, non-deterministic algorithms, NP-hard graph problems and scheduling problems.

List of Practical:

1. Store data of students using hashing function for roll number and implement linear probing using chaining without replacement and chaining with replacement algorithm.

2. a) The internship is offered to students based on rank obtained in the second year of graduation. Create a suitable non-linear data structure to identify the next topper student for internship. (Create max-heap).
b) Sort the student data in ascending order of grades.

3.Divide and Conquer:
Implement a program to find minimum and maximum elements from a given list using Divide and Conquer strategy.

4.Greedy approach:
A business house has several offices in different countries; they want to lease phone lines to connect them with each other and the phone company charges different rent to connect different pairs of cities. Business house wants to connect all its offices with a minimum total cost. Represent using appropriate data structure.
Apply Prim’s and Kruskal’s algorithm to find the minimum total cost.

5.Greedy and dynamic approach:
a) Use the map of the area around the college as the graph. Identify the prominent landmarks as nodes and find minimum distance to various landmarks from the college as the source. Represent this graph using an adjacency matrix. Find the shortest path using Dijkstra’s algorithm.
b) Write a program to implement the Bellman-Ford algorithm to find the shortest path from a single source to all other nodes in a graph with negative edge weights. Verify its results for a sample graph and compare it with Dijkstra’s algorithm.

6.Backtracking:
a) Solve Graph Coloring problem using backtracking approaches.
b) N-Queens Problem: Write a recursive program to find the solution of placing N- queens on a chess board so that no queen takes each other.

7.Branch and Bound:
a)Write a program to solve the travelling salesman problem. Print the path and the cost.
b) Implement branch and bound for the 0/1 Knapsack problem.

8.Distinguish NP-Complete problems (in NP and hardest within NP, e.g., set cover) from NP-Hard problems (at least as hard as NP but not necessarily in NP), and show how distributed join ordering becomes NP-Hard due to factors like data fragmentation, network cost, and the huge number of possible join plans.

Sample List of Project areas:
1. Path Planning Algorithms for Smart Campus Navigation
2. Optimized Scheduling Using Priority Queues & Heaps
3.Compression Using B-Trees
4.Collision-resistant hash table visualizer using linear probing, quadratic probing, rehashing, and chaining.
5.Mini-DBMS implementing transactions, locking, deadlock detection, and log-based recovery
6.MongoDB-based any analytics system comparing SQL vs NoSQL performance

Text Books:

1. Horowitz and Sahani, "Fundamentals of computer Algorithms", Galgotia. ISBN 81-7371-612-9.
2. Data Structures and Algorithms in C++" by Goodrich, Tamassia, Goldwasser
3. Thomas H Cormen and Charles E.L Leiserson, "Introduction to Algorithm" PHI, ISBN:81-203-2141-3.

Reference Books:

1. Brassard & Bratley, ―Fundamentals of Algorithmics‖, Prentice Hall India/Pearson Education, ISBN 13-9788120311312
2. AnanyLevitin, "Introduction to the Design & Analysis of Algorithm ",Pearson ISBN 81- 7758-835-4
3. "Data Structures using C and C++" by Yedidyah Langsam, Moshe J Augenstein, Aron M Tenenbaum

e Learning Resources:

1.NPTEL course- Data Structures And Algorithms, IIT Delhi by Prof. Naveen Garg, https://nptel.ac.in/courses/106102064

2. NPTEL course -Programming, Data Structures And Algorithms, IIT Madras ,Prof. Hema A Murthy, Prof. Shankar Balachandran, Prof. N. S. Narayanaswamy,

https://archive.nptel.ac.in/courses/106106127/

3.NPTEL course -Programming, Data Structures And Algorithms Using Python, Prof. Madhavan Mukund, Chennai Mathematical Institute, https://archive.nptel.ac.in/courses/106106145/

4. Infosys Springboard course- Data Structures and Algorithms, https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01384203240484864010470_shared/overview

5.Springboard course -Algorithm Design Paradigms, https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01329471165341696025446shared/overview

Course Outcomes:

The student will be able to –

CO1: After completion of the course, student will be able to Explore various advanced data structures and their operations.

CO2: Apply the principles of hashing and collision resolution techniques and implement hash tables to achieve efficient data storage and retrieval.

CO3: Analyze greedy algorithms and dynamic programming algorithms for various optimization problems.

CO4: Comprehend the principles and techniques of the backtracking and Branch and Bound Method as a problem-solving paradigm.

CO-PO Mapping Matrix:

CO	**Program Outcomes (PO)**											**PSO**		
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	1	2	2	-	-	-	-	-	-	-	3	1
CO2	3	2	2	2	3	-	-	-	-	-	-	-	3	2
CO3	3	3	2	3	2	-	-	-	-	-	-	-	2	1
CO4	3	3	2	3	2	-	-	-	-	-	-	-	2	1
Average	3	2.5	1.75	2.5	2.25	-	-	-	-	-	-	-	2.5	1.25

Justification for Mapping for all Cos

CO1 Justification

“Explore various advanced data structures and their operations.”

- **PO1 (3):** Requires strong knowledge of computing fundamentals for trees, heaps, tries, B/B+ trees.

- **PO2 (2):** Involves selecting appropriate data structures for problem solving.
- **PO3 (1):** Low mapping—design is not deeply emphasized.
- **PO4 (2):** Students perform experimentation (e.g., implementing AVL, heaps).
- **PO5 (2):** Use of programming tools & debugging.
- **PO10 (1):** Documentation of operations and outputs.
- **PSO1 (3):** Strong mapping—data structure knowledge is fundamental for IoT system implementation.
- **PSO2 (1):** Basic mapping—hash trees appear in Blockchain but lightly.

CO2 Justification

“Apply hashing and collision resolution techniques and implement hash tables.”

- **PO1 (3):** Core computational knowledge in hashing.
- **PO2 (2):** Problem analysis to choose appropriate hashing strategy.
- **PO3 (2):** Designing hash table solutions.
- **PO4 (2):** Implementing and evaluating hash table performance.
- **PO5 (3):** Practical implementation and debugging.
- **PO10 (1):** Documenting implementation constraints.
- **PSO1 (3):** IoT systems need efficient lookup structures (sensor IDs, routing tables).
- **PSO2 (2):** Hashing underlies Blockchain (hash functions, chaining).

CO3 Justification

“Analyze greedy algorithms and dynamic programming algorithms for optimization problems.”

- **PO1 (3):** Requires mathematical foundations for algorithm analysis.
- **PO2 (3):** Identifying optimization needs and constraints.
- **PO3 (2):** Designing solutions using DP and greedy paradigms.
- **PO4 (3):** Evaluating correctness and performance.
- **PO5 (2):** Coding DP and greedy algorithms.
- **PO10 (1):** Presenting results/interpretation.
- **PSO1 (2):** Optimization algorithms are often used in IoT (routing, scheduling).
- **PSO2 (1):** Some DP concepts relate to cryptographic optimization.

CO4 Justification

“Comprehend and apply backtracking & Branch-and-Bound techniques.”

- **PO1 (3):** Strong computational basis.
- **PO2 (3):** Understanding constraints & pruning.
- **PO3 (2):** Design of combinatorial solutions (N-queens, TSP).
- **PO4 (3):** Evaluate time complexity and optimality.
- **PO5 (2):** Implementation skills.
- **PO10 (1):** Communicating results and complexity.
- **PSO1 (2):** Useful in IoT for scheduling, routing constraints.
- **PSO2 (1):** Optimization also has links to secure computation problems.

CB2010: Machine Learning

Credits: 3 **Teaching Scheme**: Theory: 2 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

Linear Algebra and Calculus, Probability Basics

Course Objectives:

1. To provide students with a solid foundation in machine learning.
2. To enable students to understand regression analysis and regression models.
3. To familiarize students with classification techniques and its performance.
4. To provide knowledge of unsupervised learning techniques.
5. To introduce the fundamentals of artificial neural networks and optimization algorithms.
6. To explore advanced machine learning topics.

Relevance of this course:

Machine Learning enables computers to learn from data and make predictions or decisions automatically, forming the core of modern technologies such as AI, data analytics, healthcare systems, recommendation engines, and intelligent automation.

SECTION I

Unit I: Introduction to Machine Learning (4 Hours)

Defining and understanding Machine Learning, Types of machine learning, Model definition, Parametric and non-parametric modeling, Generalization: Concept of Training, Validation, Testing, Overfitting and Under fitting as applied to models. Concept of Bias and Variance and its importance in machine learning. Feature Engineering and Dimensionality reduction. Applications of Machine Learning.

Unit II: Supervised Learning: Regression, Classification (10 Hours)

Regression: Linear Models: Least Square method and Polynomial regression, Univariate Regression, Multivariate Linear Regression, Logistic regression, Principal Component Analysis, Error function used in regression

Binary and Multiclass Classification: Assessing Classification Performance, Handling more than two classes, Multiclass Classification-One vs One, One vs Rest Linear Models:Linear Discriminant Analysis, Bayesian Classification, Support Vector Machines (SVM), Decision Trees, Random Forest classifiers, Minority Class, Impurity Measures – Gini Index and Entropy, Best Split.

SECTION II

UNIT III: Unsupervised Learning (7 Hours)

Distance Based Models: Distance based clustering algorithms K-NN, K-means and K-medoids, Hierarchical clustering, Density-based Clustering, Rule Based Models: Rule learning for subgroup discovery, Association rules mining – Apriori Algorithm, Confidence and Support parameters.

Unit IV: Advanced Neural and Modern Learning Techniques (7 Hours)

Artificial neuron model, Concept of bias and threshold, Activation functions, Gradient descent algorithm and application of linear neuron for linear regression and classification, Multilayer perceptron (MLP) and back propagation algorithm , Reinforcement Learning, Architecture of ConvNet, Convolution Layer, Pooling Layer, and Applications of CNN's in Computer Vision, Case study using CNN, Introduction to Recurrent neural networks.

List of Practical:

1. Study of WEKA
2. Study of Google Colab and libraries used in machine learning
3. Generate a 2-D dataset and perform Linear, Multiple, and Polynomial Regression; plot Training/Test MSE
4. Implement Logistic Regression in Colab
5. Implement SVM for classification and find accuracy
6. Implement Naïve Bayes algorithm and evaluate performance metrics using WEKA API
7. Implement K-means algorithm to create clusters on given data
8. Create Association Rules for Market Basket Analysis
9. Implement and test Multi-Layer Perceptron (MLP) trained with back-propagation
10. Implement and test Convolutional Neural Network (CNN) for digit recognition
11. Real-time mini project using ML algorithms for industry/research problem

Sample List of Project areas:

1. House Price Prediction-Regression (Linear / Multiple / Ridge / Lasso)
2. Student Performance Prediction-Regression and Error Analysis
3. Stock Market Trend Forecasting-Time Series Regression
4. Email Spam Detection-Binary Classification (Logistic Regression / SVM)

Text Books:

1. EthemAlpaydin: Introduction to Machine Learning, PHI 2nd Edition2013.
2. Peter Flach: Machine Learning: The Art and Science of Algorithms thatMake Sense of Data, Cambridge University Press, Edition 2012

Reference Books:

1. C. M. Bishop: Pattern Recognition and Machine Learning, Springer 1st Edition-2013.
2. Ian H Witten, Eibe Frank, Mark A Hall: Data Mining, Practical Machine Learning Tools and Techniques, Elsevier, 3rd Edition
3. Parag Kulkarni: Reinforcement Learning and Systemic Machine Learning for Decision Making, IEEE Press, Reprint 2015.
4. Nikhil Buduma: Fundamentals of Deep Learning, O'Reilly Media, June 2017.

5. Hastie, Tibshirani, Friedman: Introduction to Statistical Machine Learning with Applications in R, Springer, 2nd Edition 2012. 6. Kevin
6. P Murphy: Machine Learning – A Probabilistic Perspective, MIT Press, August 2012

e Learning Resources:

1. Machine Learning and Pattern Recognition - Course
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_014157683278520320131/overview
3. https://onlinecourses.nptel.ac.in/noc26_cs77/preview
4. https://onlinecourses.nptel.ac.in/noc26_cs74/preview
5. https://onlinecourses.nptel.ac.in/noc26 _cs76/preview

Course Outcomes:

The student will be able to –

CO1: Understand foundational ML concepts such as generalization, bias–variance, model types, feature engineering, and dimensionality reduction.

CO2: Apply regression and classification techniques to develop and evaluate supervised machine learning models.

CO3: Analyze unlabeled data using clustering, dimensionality reduction, and association rule mining techniques.

CO4: Design and optimize neural network models and advanced machine learning architectures like MLP, CNN, RNN, and ensemble methods..

CO-PO Mapping Matrix:

CO	Program Outcomes (PO)											PSO		
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	3	2					2		3	2	
CO2	3	3	2	3	2				1	2	1	3	2	1
CO3	3	3	2	3	1					2		3	3	
CO4	3	3	3	3	3				1	2	1	3	3	1
Average	3	2.75	2	3	2	0	0	0	0.5	2	0.5	3	2.5	0.5

Justification for Mapping for all Cos

CO1 Justification

“Understand foundational machine learning concepts, including bias-variance trade-off and dimensionality reduction.”

- **PO1**-Involves applying mathematical foundations such as linear algebra, statistics, and probability required for ML theory.
- **PO2**-Requires problem analysis to understand data behaviour, sources of variance, and model generalization properties.
- **PO3**-Supports the ability to identify suitable ML approaches while considering model design constraints.
- **PO4**-Encourages selection and application of ML techniques using appropriate tools and software.
- **PO5**-Requires modern computational tools for visualizing datasets and dimensionality reduction techniques.
- **PO10**-Understanding ML fundamentals enhances the ability to communicate technical concepts clearly.
- **PSO1**-Builds strong foundations in computing concepts required for advanced data analysis.
- **PSO2**-Develops analytical thinking essential for pattern recognition and data interpretation.

CO2 Justification

“Apply regression and classification techniques for predictive modeling.”

- **PO1**-Requires strong mathematical and statistical knowledge for implementing ML algorithms.
- **PO2**-Involves analysing problem statements and datasets to design effective predictive models.
- **PO3**-Requires development of models using engineering principles and structured workflows.
- **PO4**-Uses ML libraries, tools, and evaluation metrics for model testing and validation.
- **PO5**-Makes use of modern software tools (Python, scikit-learn, Jupyter) to develop solutions.
- **PO9**-Often performed as team-based mini projects involving collaborative development.
- **PO10**-Requires documentation and presentation of model findings and performance results.
- **PO11**-Involves planning, training and evaluating ML models within given constraints.
- **PSO1**-Applies programming knowledge to solve real-world predictive problems.
- **PSO2**-Enhances analytical ability to classify and predict outcomes.
- **PSO3**-Addresses domain-specific applications of ML such as finance, health, or business prediction.

CO3 Justification

“Analyze unlabelled data using clustering and dimensionality reduction techniques.”

- **PO1**-Requires mathematical foundations for similarity measures and distance metrics.
- **PO2**-Involves identifying inherent patterns in unlabeled data by analyzing characteristics.
- **PO3**-Supports modelling decisions such as choosing appropriate clustering or DR algorithms.

- **PO4**-Uses tools such as scikit-learn, Python libraries, and visualization frameworks.
- **PO5**-Makes use of modern software to implement and validate clustering results.
- **PO10**-Requires clear presentation of clustering outputs and data insights.
- **PSO1**-Applies core computing skills for designing clustering models.
- **PSO2**-Strengthens analytical and problem-solving skills in unsupervised learning.
- **PSO3**-Supports real-world data segmentation use-cases like customer clustering.

CO4 Justification

“**Design and optimize artificial neural networks for regression and classification.”**

- **PO1**-Requires deep understanding of mathematical concepts like gradients and optimization.
- **PO2**-Involves analyzing complex problems and selecting suitable neural architectures.
- **PO3**-Requires designing ANN models, tuning hyperparameters, and validating performance.
- **PO4**-Uses advanced ML libraries such as TensorFlow, Keras, or PyTorch.
- **PO5**-Relies heavily on modern tools, GPU computing, and software frameworks.
- **PO9**-Neural network projects are often implemented collaboratively.
- **PO10-**Requires effective communication of model architecture, results, and insights.
- **PO11-**Involves planning, scheduling, and executing ML model development as a project.
- **PSO1**-Strengthens advanced programming and computing skills required for ANN design.
- **PSO2**-Enhances analytical capability in solving complex nonlinear problems.
- **PSO3**-Supports domain-specific applications like image classification, forecasting, and NLP.

CB2011: Database Management Systems

Credits: 3 **Teaching Scheme**: Theory: 2 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

Programming fundamentals, Data structures

Course Objectives:

1. Understand fundamental concepts of databases, including DBMS architecture, data models, and the process of mapping ER diagrams to relational schemas.
2. Learn to design efficient, well-normalized relational database schemas and implement data definition and data manipulation operations using SQL.
3. Learn advanced DBMS concepts such as transaction management, concurrency control, indexing techniques, and emerging NoSQL database systems.
4. Explore modern database systems focusing on security, intelligence, and distributed architectures, with applications in IoT, AI-driven systems, and compliance-oriented environments.

Relevance of this course:

DBMS provides the foundation for efficient storage, retrieval, and management of data, which is essential for building reliable software applications, supporting data security, ensuring data consistency, and enabling data-driven decision-making in modern information systems.

SECTION I

Unit I: Database Concepts, Data Modeling & Design (7 Hours)

Introduction to Databases and DBMS, Database Users & Applications, DBMS Architecture: Centralized, Client–Server, Distributed, Role of DBMS in IoT, Cybersecurity, Blockchain, and AI systems, Data Models: Entity–Relationship (ER) Model; Relational Data Model, ER Diagram components: entities, attributes, relationships, constraints, Mapping ER Model to Relational Schema, Keys: Super, Candidate, Primary, Foreign, Integrity Constraints: Domain, Entity, Referential, ER diagram Case studies.

Unit II: Relational Database Design & SQL Fundamentals (7 Hours)

Relational Database Design Principles, Functional Dependencies, Normalization(1NF-BCNF),Introduction to SQL, DDL,DML, (DCL – overview), Constraints in SQL (Primary, Foreign Key, CHECK, UNIQUE) hands-on practice using SQL tools.

SECTION II

UNIT III: Advanced DBMS Concepts & Non-Relational Databases (7 Hours)

Advanced SQL-Joins (INNER, OUTER), Subqueries and Views, Window Functions (ROW_NUMBER, RANK, aggregation over windows), Indexing and Query Processing, Types of indexes, Query execution plans; Transaction Management, ACID properties, Concurrency control, Locks and isolation levels, Backup, Recovery, and Replication; Introduction to NoSQL Databases: Key–Value stores, Document databases, Use of NoSQL in IoT and high-velocity data systems.

Unit IV: Secure, Intelligent & Distributed Databases (7 Hours)

Time-Series and Edge Databases for IoT Sensor Data, Vector Databases and Semantic Search, Embeddings and similarity search, Retrieval-Augmented Generation (RAG) overview; Databases in AI/ML Pipelines - Feature stores, Data versioning, Database Security- Authentication and authorization, Encryption at rest and in transit, Data Privacy & Compliance - Indian Digital Personal Data Protection (DPDP) Act, Data residency considerations; Blockchain Data Storage ConceptsSecure device-to-database pipelines and OT, A data ingestion (overview)

List of Practical:

1. Install PostgreSQL/MySQL; verify access. Design and implement a relational database for a real-world application (e.g., Student Management / E-commerce / Smart Device Inventory). Identify entities, attributes, and relationships, apply normalization rules, and create tables using appropriate data types and constraints in MySQL.
2. Implement Data Definition Language (DDL) commands to create, alter, and drop tables. Enforce data integrity using PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK, NOT NULL, and DEFAULT constraints and analyze their impact on data consistency.

3. Write at least 15 SQL queries using aggregate functions, GROUP BY, HAVING, nested queries, and subqueries to perform analytical operations such as summaries, counts, averages, and conditional data extraction.

4. Implement and analyze INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN operations across multiple tables. Map SQL queries to equivalent relational algebra expressions to understand query execution logic.
5. Create and use views for data abstraction and security. Implement indexes to optimize query performance and analyze execution time before and after indexing. Create simple stored procedures or functions for reusable database operations.
6. Demonstrate transaction processing using COMMIT, ROLLBACK, and SAVEPOINT. Simulate concurrent transactions and analyze anomalies such as dirty reads and lost updates. Explain how ACID properties ensure database reliability.
7. Develop an application using a programming language (Python/Java) to establish connectivity with a MySQL database. Implement secure CRUD operations using parameterized queries and demonstrate database navigation using result sets. Implement database security by creating users and roles. Assign and revoke privileges using GRANT and REVOKE commands.
8. Implement aggregation and indexing with suitable example using Mongo DB.
9. Implement Retrieval-Augmented Generation (RAG) Using Vector Databases and Pretrained

Large Language Models

10. Mini Project

I.Relational Databases - SQL Queries, Joins, Views, Indexes, Normalization (1NF, 2NF, 3NF, BCNF), Transactions & ACID Properties, Triggers (Row-level & Statement-level), Stored Procedures & Functions, Reporting & Analytics Queries.

Samples topics:

1. E-Commerce Order Management System
2. Hospital Management System
3. Bank Management System
4. Online Examination & Student Performance System
5. Inventory & Supply Chain Management System

II. NoSQL / MongoDB-Based Projects - MongoDB CRUD operations, Indexing strategies (single, compound, text, geospatial), Aggregation framework & analytics, Schema design (nested documents, arrays, flexible schemas), Replication & sharding (high availability and scalability), Time-series, real-time data, and multimedia handling, Data security, privacy, and compliance. Samples topics:

1. E-Commerce Product Catalog & Recommendation System
2. Real-Time Chat & Social Media Analytics Platform
3. Healthcare Patient Records & Clinical Analytics
4. Log Management & Real-Time Analytics System

Course Project (CP):

1. Secure IoT Sensor Data Storage & Analytics Platform
2. Smart Water Metering DB with Anomaly Detection
3. Blockchain-based Certificate Storage System
4. Secure Healthcare Records DB (DPDP-aware)
5. EV Charging Station Transaction Database
6. Smart Agriculture Time-Series DB
7. Cybersecurity Log Analytics Platform
8. Vector-DB-powered Academic Search Engine
9. Smart Attendance System using IoT & DB
10. Supply Chain Traceability Database
11. Secure E-Governance Data Portal
12. Fraud Detection DB using Transaction Data
13. SMS-based Remote Server Monitoring System
14. Voice-Based Transport Enquiry System
15. Cybersecurity Threat Intelligence Platform Management System.
16. Artificial Intelligence Learning Model Data Handling Management System.
17. Stock Market Simulation Management System

Text Books:

1. Abraham Silberschatz, Henry Korth, S.Sudarshan," Database System concepts",5th Edition, McGraw Hill International Edition.
2. Elmasri R.,Navathe S.," Fundamentals of Database Systems", 4*Edition, Pearson Education, 2003, ISBN 8129702282.
3. Pramod J. Sadalage and Martin Fowler, “NoSQL Distilled”, First Edition, Addison Wesley, ISBN10:0321826620,ISBN-13: 978-0321826626

Reference Books:

1. Managing and Using MySQL", Reese G., Yarger R., King T., Williums H, 2nd Edition, Shroff Publishers and Distributors Pvt.Ltd., ISBN81 -7366 - 465–X
2. Kristina Chodorow, "Mongo DB: The Definitive Guide", 2nd Edition, O'Reilly, ISBN-10: 1449344682, ISBN-13: 9781449344689
3. Ramkrishna R., Gehrke J., "Database Management Systems", 3rd Edition, McGraw-Hill, 2003, ISBN 0-07-123151 –X.
4. C.J. Date, "An Introduction to Database Systems", Addison-Wesley/ Pearson Education, ISBN:0201144719
5. Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN81-7808-861-4

e Learning Resources:

1. SQL - https://data-flair.training/courses/free-sql-course-hindi/
2. Book Slides - https://www.db-book.com/slides-dir/index.html
3. NPTEL - https://nptel.ac.in/courses/106105175

Course Outcomes:

The student will be able to –

CO1: Explain database fundamentals, architectures, data models, and ER-to relational mapping.

CO2: Design normalized relational databases and implement SQL operations.

CO3: Apply advanced DBMS concepts including transactions, indexing, and NoSQL databases.

CO4: Analyze modern secure, intelligent, and distributed databases for IoT, AI, and compliance-driven systems.

CO-PO Mapping Matrix:

CO	Program Outcomes (PO)												PSO	
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	3	2	1	–	–	–	–	–	–	–	–	1	2	1
CO2	3	3	3	2	2	–	–	–	1	–	–	1	3	1
CO3	3	3	2	3	3	1	–	–	1	–	–	2	3	2
CO4	2	3	3	3	3	2	1	3	1	1	–	2	3	3
Average	2.75	2.75	2.25	2	2	0.75	0.25	0.75	0.75	0.25	0	1.5	2.75	1.75

Justification for Mapping for all Cos

CO1 Justification:

“Explain database fundamentals, architectures, data models, and ER-to-relational mapping”

CO–PO Justification:
PO1: By teaching students core DBMS concepts, data models, and ER diagrams. It also enhances problem analysis skills
PO2 : As students identify and map entities, relationships, and constraints. Students apply basic solution design skills
PO3 : While creating relational schemas from ER diagrams. Additionally, CO1 promotes lifelong learning
PO12: Through exploration of case studies and independent study of DBMS principles.

CO–PSO Justification:
CO1 contributes to **PSO1** by providing foundational knowledge for designing IoT-centric data management solutions. It has limited alignment with **PSO2**, as it introduces basic concepts of data security and compliance.

CO2 Justification:

“Design normalized relational databases and implement SQL operations”

CO–PO Justification:
PO1 PO2 PO3 : Strongly aligns with as students design normalized schemas and implement SQL queries.
PO4 : It supports (investigation of complex problems) as students analyze functional dependencies and normalization forms.
PO5 : It is addressed through hands-on SQL labs.
PO9 : It also fosters **teamwork skills**
PO12 : During collaborative exercises and encourages **self-learning and continuous improvement**

CO–PSO Justification:
CO2 strongly maps to **PSO1** since it equips students with practical skills for designing and implementing IoT and AI data-driven systems. It has a minimal link to **PSO2**, as SQL implementation involves limited security or compliance considerations.

CO3 Justification:

“Apply advanced DBMS concepts including transactions, indexing, and NoSQL databases”

CO–PO Justification:

PO1: Develops and problem analysis skills .
PO2 : by exposing students to transaction management, concurrency, and indexing.
PO3 : It addresses solution design.
PO4 : By teaching students to optimize database queries and manage distributed or high-velocity data.
PO5 : It is achieved through NoSQL and advanced SQL practice. Students gain awareness of societal impacts.
PO6 : Through discussions on high-velocity data systems, and teamwork.
PO9: It is practiced in lab exercises. CO3 also reinforces lifelong learning.
PO12 : as students explore emerging database technologies.

CO–PSO Justification:

CO3 aligns strongly with **PSO1**, enabling design and implementation of scalable IoT and AI data systems. It also partially supports **PSO2**, as students learn concepts relevant to secure data management and compliant data handling.

CO4 Justification:

“Analyze modern secure, intelligent, and distributed databases for IoT, AI, and compliance-driven systems”

CO–PO Justification:

PO2 PO3 PO4: Strong alignment with because students analyze time-series databases, vector databases, RAG workflows, and feature stores.
PO5 : It addresses through hands-on exposure to distributed and AI-integrated databases.
PO6 PO8: They are emphasized through discussions on data privacy, DPDP Act compliance, and blockchain storage.
PO9 PO10 : During project work and presentations, while encouraging
PO12 : As students explore emerging database technologies and secure data solutions.

CO–PSO Justification:

CO4 strongly maps to **PSO1**, as it equips students to design IoT and AI pipelines using distributed and feature-store databases. It also strongly maps to **PSO2**, since students apply data security, encryption, privacy, and compliance principles in modern database systems.

CB2012: Computer Networks

Credits: 3 **Teaching Scheme**: Theory: 2 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

Knowledge of data communication

Course Objectives:

1. To introduce the fundamental concepts of computer networks,switching techniques, and reference models.
2. To explain the functions and protocols of different network layers, with emphasis on physical, data link, network, transport, and application layers.
3. To develop the ability to understand IP addressing, routing, and subnetting, and apply these concepts using networking tools such as Cisco Packet Tracer.
4. To familiarize students with transport and application layer protocols, enabling analysis of reliable and unreliable data communication in real-world networked applications.

Relevance of this course:

The Computer Networks course explains how data is transmitted between devices, forming the backbone of the internet, communication systems, cloud computing, and secure networked applications used in today's digital world.

SECTION I

Unit I: Introduction to Computer Network **(7 Hours)**

Switching, Circuit-switched Networks: Three Phases, Efficiency, Delay, Packet switching: Datagram networks, Virtual circuit networks, Brief introduction of Digital Subscriber Line: ADSL, HDSL, SDSL, VDSL (DMT), Cable modem.

Unit II: Physical Layer **(7 Hours)**

Communication rules and the Internet Network Architectures: Client-Server; Peer To Peer; Network as a Platform, Network Topologies, OSI Model, TCP/IP protocol suite; Layer Details, Addressing: Physical &logical Addresses, Port Addresses, Specific Addresses. Connecting devices: Hubs (Passive, active, Intelligent), Switches (Layer-2, Layer-3 and Managed), Bridges, Routers, Gateway

SECTION II

UNIT III: Data Link Layer and Network Layer **(7 Hours)**

Data Link Layer Protocols, MAC address, Address Resolution Protocol, Hop to Hop delivery, Network Layer Protocols, Routing- Static and Dynamic, Routers, configuring a Router using Cisco Packet Tracer. IP Addressing: IPv4 Network Addresses, IPv4 Datagram Format, Fragmentation and Reassembly. IPv6 Network Addresses, Connectivity Verification. Subnetting IP Networks: Subnetting an IPv4 Network, designing subnetting using Cisco Packet Tracer

Unit IV: Transport Layer and Application Layer (7 Hours)

Transport Layer: Transport Layer Protocols, Role of transport layer, Responsibilities of Transport layer, Transport layer reliability. TCP and UDP: TCP communication Process, Socket Programming, Reliability and flow control, Port Addressing, Multiplexing/ Demultiplexing. UDP (Segment Format, Checksum),applications of TCP and UDP. Application Layer: Application Layer Protocols, Application layer protocols interaction with end-user applications, Presentation and Session layers. Well-Known Application Protocols and Services, Domain Name System (DNS), HyperText Transfer Protocol (HTTP).

List of Practical:

1. Study of Network Types and Switching Techniques Analysis
2. Creating a Computer Lab designing with PC's and Switch Configuration using Cisco Packet Tracer
3. Inter-LAN communication Using routers and static routing
4. To analyze efficiency and delay in a circuit-switched network for given bandwidth and data size in Cisco Packet Tracer or Calculator.
5. To configure static routing between multiple networks using routers in Cisco Packet Tracer.
6. To simulate a datagram-based packet-switched network and observe independent routing of packets.
7. To study DSL technologies (ADSL, HDSL, SDSL, VDSL-DMT) and Cable Modem architecture using diagrams and data flow
8. Design and Build a Small Business Network using Packet Tracer.
9. To configure IPv4 addresses and verify connectivity using ping and tracer commands.
10. To demonstrate MAC addressing and ARP protocol using Cisco Packet Tracer.
11. To configure and analyze dynamic routing protocols (RIP / OSPF) using Cisco Packet Tracer.
12. To design an IPv4 sub netted network and implement it using Cisco Packet Tracer.

Sample List of Project areas:

1. ARQ Protocol Simulator
2. Subnetting and Routing Design
3. TCP/UDP Traffic Analysis using Wireshark

Text Books:

1. Andrew S. Tanenbaum ,"Computer Networks", Pearson, Sixth Edition
2. Stallings William., "Data and Computer Communications", Sixth Edition, Prentice Hall of India.
3. Fourouzan B., "Data Communications and Networking", 5th edition, McGraw- Hill Publications

Reference Books:

1. CCNA Routing and Switching 200-125 Official Cert. Guide Library
2. Cisco CCNA Command Guide- An Introductory Guide for Complete Beginners

e Learning Resources:
CISCO CCNA Certification

Course Outcomes:

The student will be able to –

CO1: Explain the fundamental concepts of computer networks, including network architectures, switching techniques, and reference models.

CO2: Describe and illustrate the functions and protocols of the physical, data link, network, transport, and application layers.

CO3: Apply IP addressing, routing, and subnetting concepts in the design and configuration of networks using Cisco Packet Tracer.

CO4: Analyse transport and application layer protocols to distinguish reliable and unreliable data communication in real-world applications.

CO-PO Mapping Matrix:

CO	Program Outcomes (PO)											PSO		
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO12	PSO1	PSO2
CO1	3	2	-	-	-	-	-	-	-	-	-	1	2	-
CO2	3	3	2	-	-	-	-	-	-	-	-	1	3	-
CO3	2	3	3	2	3	-	-	-	-	-	-	2	3	2
CO4	2	3	2	2	-	-	-	-	-	-	-	3	2	3

Justification for Mapping for all Cos

CO1 Justification

"Explain the fundamental concepts of computer networks, including network architectures, switching techniques, and reference models"

- **PO1-** Strengthens core engineering knowledge by introducing foundational networking concepts, architectures, switching techniques, and reference models.

- **PO2 -** Understanding basic networking concepts enables students to analyze simple networking problems and identify appropriate models and techniques.
- **PO12 -** Exposure to standard networking principles motivates students to continuously update their knowledge as networking technologies evolve.
- **PSO1 -** Directly supports program-specific knowledge in computer networks and communication fundamentals.

CO2 Justification

"Describe and illustrate the functions and protocols of the physical, data link, network, transport, and application layers."

- **PO1 -** Depends students' understanding of engineering principles related to layered network architectures and protocol operations.
- **PO2 -** Layer-wise protocol knowledge helps students analyse communication issues and identify faults within specific network layers.
- **PO3 -** Understanding protocol functions assists students in designing basic networking solutions using appropriate layers and protocols.
- **PO12 -** Familiarity with standardized protocols encourages continuous learning as new protocols and updates emerge.
- **PSO1 -** Strongly aligns with program-specific outcomes related to protocol analysis and layered network design.

CO3 Justification

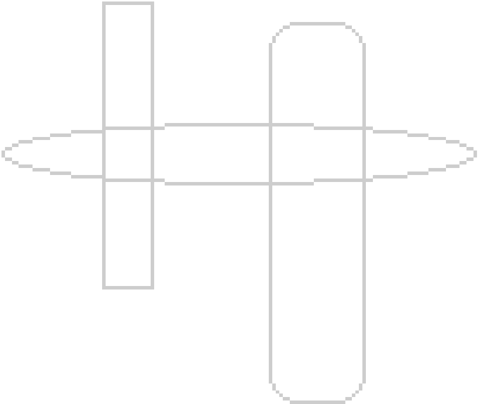
"Apply IP addressing, routing, and subnetting concepts in the design and configuration of networks using Cisco Packet Tracer."

- **PO1 -** Applies theoretical networking knowledge to practical IP addressing and routing scenarios.
- **PO2 -** Subnetting and routing require analysis of network requirements and constraints to derive efficient solutions.
- **PO3 -** Students design and configure networks by creating addressing schemes and routing solutions.
- **PO4 -**Network configuration tasks involve investigating connectivity issues and performance considerations.
- **PO5 -** Emphasizes the use of modern networking tools such as Cisco Packet Tracer for simulation and configuration.
- **PO12 -** Hands-on tool-based learning prepares students to adapt to evolving networking platforms and technologies.
- **PSO1 -** Reinforces program-specific competence in network design and IP-based communication.
- **PSO2 -** Practical subnetting and routing exercises enhance students' applied skills in configuring real-world networks.

CO4 Justification

“ Analyse transport and application layer protocols to distinguish reliable and unreliable data communication in real-world applications.”

- **PO1**-Builds on engineering principles related to reliable and unreliable data transmission mechanisms.
- **PO2 -**Analysis of TCP and UDP enables students to evaluate communication requirements and select suitable protocols.
- **PO3 -**Understanding protocol behaviour assists in designing application-specific communication solutions.
- **PO4-** Protocol analysis involves investigating performance, reliability, and flow control issues.
- **PO12 -** Rapid evolution of application protocols encourages continuous learning and professional development.
- **PSO1 -** Supports advanced understanding of transport and application layer concepts.
- **PSO2 -** Analysis of real-world protocols prepares students to implement and troubleshoot networked applications.



MM0502: Software Project Management (CBM002)

Credits: 3 **Teaching Scheme**: Theory: 2 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

Programming Knowledge and Problem Solving

Course Objectives:

1. To understand the nature of software projects, stakeholders, feasibility analysis, and project initiation activities.
2. To apply software project planning techniques including estimation, scheduling, and resource management.
3. To understand and apply Agile and DevOps practices for managing modern software projects.
4. To manage software project risks, quality, changes, and monitoring activities effectively.

Relevance of this course:

Software Project Management is highly relevant as it equips students with the skills to plan, execute, monitor, and deliver software projects successfully. The course helps learners understand project planning, cost and time estimation, risk management, team coordination, and quality assurance, which are essential in the software industry. It prepares students to handle real-world project challenges, work effectively in teams, meet deadlines, and deliver projects within budget, making them industry-ready and improving their employability and leadership potential

SECTION I

Unit I: Introduction to Software Project Management (7 Hours)

Characteristics of Software Projects, Software Crisis and Project Management Challenges, Project success factors. Software Development Life Cycle Models & Frameworks Project Management: Project Definition, Project Initiation, Categorization and Types of Software Projects, Project Planning objectives, TELOS feasibility analysis, Stakeholder Identification, Project Charter, Overview of AI-assisted Project Planning tools, Case discussion on successful and failed software projects.

Unit II: Project Planning and Effort Estimation (7 Hours)

Software size estimation using LOC and Function Points, effort and cost estimation models including Basic COCOMO, project scheduling techniques such as Gantt chart, PERT and CPM, resource allocation and workload balancing, budgeting and cost control, use of AI tools for estimation and schedule optimization. Hands-on estimation and scheduling exercise

SECTION II

UNIT III: Agile, Devops, And Modern Practices (7 Hours)

Agile manifesto and principles, Scrum framework roles events and artifacts, user stories backlog management and sprint planning, agile estimation and velocity, DevOps concepts and CI/CD overview, project tracking tools(such as Jira and GitHub Projects), role of AI in agile analytics and sprint prediction. Sprint planning and backlog preparation simulation.

Unit IV: Risk, Quality, And Project Control (7 Hours)

Project risk management process, risk identification analysis and mitigation, software quality management and metrics, configuration and change management, project monitoring control and reporting, communication management ethics and project closure, AI-based risk prediction and decision support. Risk assessment and mitigation plan preparation.

List of Tutorials:

- Tutorial 1: Prepare a Feasibility Study Report using TELOS & create a Project Charter (Tools: Miro/Docs).
- Tutorial 2: Create SRS document and Develop Requirement Models – Use Case Diagram, Context Diagram, Basic Flowchart (Tools: Draw.io).
- Tutorial 3: Create Low-Fidelity Wireframes for the proposed system (Tools: Figma/Balsamiq).
- Tutorial 4: Perform Software Estimation using Function Point Analysis and COCOMO (Tools: Calculators/Excel).
- Tutorial 5: Prepare a complete Gantt chart & PERT/CPM network (Tools: MS Project/GanttProject).
- Tutorial 6: Develop a Risk Register with probability–impact scores and mitigation plans (Tools: Excel).
- Tutorial 7: Agile Simulation – Write Agile user stories , Create product backlog, sprint backlog & burndown chart (Tools: Jira/Trello).
- Tutorial 8: Sprint planning and velocity calculation exercise
- Tutorial 9: Study and review of an AI-based project management tool
- Tutorial 10: Final Prototype + Project Monitoring Dashboard (Tools: Figma + Jira + GitHub).

Sample List of Project areas:

- Hospital Management System – Feasibility, Estimation & Scheduling, Quality planning
- Library Automation System – Project Charter & Resource Planning
- Agile Development of Food Delivery App
- Scrum-Based Online Shopping Platform
- Fitness Tracking App using Agile Methodology
- Change Management in Large-Scale ERP Systems

Text Books:

1. Software Project Management, 5th Edition, Bob Hughes, Mike Cotterel, Rajib Mall, McGraw-Hill, 2018
2. Software Engineering: A Practitioner's Approach – Roger Pressman.
3. Agile Project Management with Scrum – Ken Schwaber.

Reference Books:

1. Managing the Software Process – Watts Humphrey.
2. Software Engineering – Ian Sommerville.
3. Project Planning, Scheduling and Control, 5th Edition, James P Lewis, McGraw Hill, 2011
4. Software Project Management in Practise, Pankaj Jalote, Pearson Education, 2002
5. Agile Software Development, Principles, Patterns and Practices, 1st Edition, Robert C. Martin, Pearson (Prentice Hall), 2002

e Learning Resources:

1. Coursera – Software Project Management Specialization (University of Alberta).
2. edX – IT Project Management (University of Washington).
3. NPTEL – Software Project Management (Prof. IIT Kharagpur).
4. FutureLearn – Digital Project Management Basics.
5. YouTube – Free tutorials on MS Project, Jira, Figma.
6. Google Project Management Certificate – Free audit option.

Course Outcomes:

The student will be able to –

CO1: Explain software project characteristics, life cycle models, stakeholder roles, and feasibility aspects.

CO2: Perform software size, effort, cost estimation, and project scheduling using standard techniques.

CO3: Apply Agile and DevOps concepts to plan, execute, and track software projects.

CO4: Analyze project risks, quality issues, and control mechanisms for effective project execution.

CO-PO Mapping Matrix:

CO	Program Outcomes (PO)												PSO	
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	2	2	1	1	1			1	1	1		1		3
CO2	1	2	3	2	3	1			2	2	1		3	1
CO3	1	3	1	3	3		1	2				2	2	
CO4		1	3		3	2		2	3	3		2		3

- **3** = Strong/Direct mapping (High Relevance)
- **2** = Moderate mapping
- **1** = Low mapping / indirect relevance
- Blank = no meaningful mapping

Justification for Mapping for all Cos

CO1 Justification

“Fundamentals of software project management, SDLC models, project initiation processes, and feasibility analysis.”

PO Mapping Logic:
PO1: fundamental theoretical knowledge of SDLC models and software project management concepts(2)
PO2: Requires analytical comparison of feasibility assessment methods and life cycle models (2)
PO3: Supports basic understanding of design implications but not detailed system design(1)
PO4: Involves elementary investigation while examining project initiation and feasibility processes(1)
PO5: Introduces limited use of documentation and modeling tools in early project stages(1)
PO8: Encourages professional approach to documenting and presenting initiation-level project information(1)
PO9: Involves cooperative work during stakeholder identification and requirements discussion(1)
PO10: Requires communication of concepts, feasibility outcomes, and project documentation(1)
PO12: Promotes lifelong learning due to evolving software project management practices and models(1)

PSO Mapping Logic:
PSO2 – Strong relevance as provides strong conceptual foundation required for software engineering domains (3)

CO2 Justification

“Project planning techniques including scope definition, WBS, scheduling, estimation models, cost and risk management in software projects.”

PO Mapping Logic:

PO1 – Uses mathematical principles in cost and effort estimation models (1)

PO2 – Strong analytical thinking required for project feasibility and estimation (2)
PO3 – Major contribution to design and structuring of project planning artifacts (3)
PO4 – Involves investigation using alternative estimation and scheduling models (2)
PO5 – Strong dependency on software tools (PERT, Gantt, Jira, MS Project) (3)
PO6 – Emphasizes societal/economic project considerations during project feasibility (1)
PO9 – Requires teamwork in scope definition and planning phases (2)
PO10 – Requires written/project plan documentation and presentation (2)
PO11 – Involves understanding sustainability and feasibility constraints (1)

PSO Mapping Logic:
PSO1 – highly relevant for structured system modeling (3)
PSO2 – partially relevant to scalable data support (1)

CO3 Justification
" Project monitoring mechanisms, configuration management practices, quality assurance models, and software metrics."
PO Mapping Logic:
PO1 – Uses theoretical knowledge for metric interpretation (1)
PO2 – Requires higher-order analytical evaluation for quality models and performance metrics (3)
PO3 – Limited contribution in design but relevant for review cycles (1)
PO4 – Strong investigative approach for analyzing deviations, trends, risks (3)
PO5 – Uses specific tools (Version control, CI/CD, Jenkins, Git, SE metrics tools) (3)
PO7 – Evaluates ethics in configuration integrity and compliance (1)
PO8 – Requires professional responsibility in quality and audit processes (2)
PO12 – Encourages continuous improvement via iterative review and metrics-based evolution (2)

PSO Mapping Logic:
PSO1 – Supports platform reliability and quality requirements for enterprise and smart systems (2)

CO4 Justification
"Agile project management methodologies and utilize modern project management tools for real-world software development."

PO Mapping Logic:
PO2 – Moderate analytical reasoning to evaluate agile frameworks (1)
PO3 – Strong design-oriented contribution through sprint planning and backlog prioritization (3)
PO5 – Heavy use of modern automation and agile tools (GitHub, Azure DevOps, Jira, Scrum boards) (3)
PO6 – Addresses end-user needs through customer-centric agile iterations (2)
PO8 – Encourages professionalism through agile roles and ceremonies (2)
PO9 – Strong teamwork relevance in Scrum, Kanban, XP practices (3)
PO10 – Verbal/written communication crucial for stand-ups, retrospectives, sprint reviews (3)
PO12 – Agile encourages continuous learning and adaptation (2)

PSO Mapping Logic:
PSO2 – Highly aligned with industry-standard agile frameworks used in real-world software engineering (3)

FF No. : 654

HS2003: From Campus To Corporate – 2

Credits:.2 **Teaching Scheme: Theory: 2 hours/Week**

Introduction to the Corporate World Understanding organizational structure and hierarchy, Work culture differences: campus vs. corporate, Employer expectations from fresh graduates, Time management and ownership in corporate settings

Professionalism: How to be a good Team Member How to be a good Mentee.

Understanding Company Financials(And Aop Budgeting): How do companies build AOP. Top down and bottom up, Understanding the funding avenues

Product Solutioning : Understand all facets, dependencies with other stakeholders and Understand a good solutioning methodology.

Product Management: Understand backlog prioritization techniques, Be able to write requirement in details.

Metrics And Measurements: Run a Case study based workshop and buildout KPIs , OKRs for a year. Understand how companies measure performance of A. Product B. Engineering C. Business, Trickle the KPI to individual Engineer level.

Understanding Ux And Ui: Compare and contrast UX with user interface. Use several industry examples technique of building a good UX.

Releasing A Product To The Market: Understand what it takes for a product to be successful with the customer from external go-to- market perspective.

Quality Understanding Beyond The Theory: Understanding Testing and Test Automation.

Customer Success Management : Understand what issues mean in the industry, Understand how to identify and prioritize issues and ways to resolve them.

Understanding Data And Data Engineering: Understand Data Science, Understand aspects of Data Science and how they are used in the industry

Technical architecture and design principles : Understand how to align product goals to technical architecture

Security and Privacy: Understand about Security Engineering Approach, Be able to prepare security test plan.

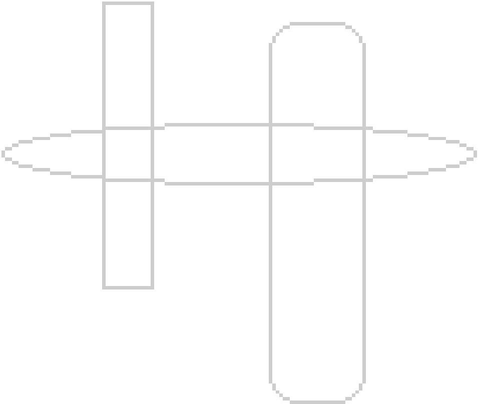
Faculty are supposed to do conduct following in the class

- Resume and LinkedIn profile workshops
- Mock interviews and GD sessions

- Role plays: workplace scenarios, conflict handling
- Business email writing exercises
- Presentation and elevator pitch sessions

Books:

1. Dale Carnegie, How to Win Friends and Influence People
2. Stephen R. Covey, 7 Habits of Highly Effective People
3. Shital Kakkar Mehra, Business Etiquette: A Guide for the Indian Professional
4. Peggy Klaus, The Hard Truth About Soft Skills



FF No. : 654

HS2004: Reasoning and Aptitude Development - 4

Credits: 1 **Teaching Scheme:** 1 Hour/Week

Unit 1: English Language

Familiarity with English Language, Ability to understand written text, spoken word and effective communication through written documents; Coverage of vocabulary to cope up with general and specific terminology, syntax and sentence structure, prevention of incorrect use leading to distortion in communication; synonyms, antonyms and contextual vocabulary, Grammar – Error identification, sentence improvement and construction, Reading Comprehension

Unit 2: Logical Ability

Objective interpretation of things, ability to perceive and interpret trends to make generalizations; ability to analyze assumptions behind an argument or statement; Deductive reasoning: Assessment of ability to synthesize information and derive conclusions - Coding deduction logic, Data Sufficiency, Directional Sense, Logical word sequence, Objective reasoning, Selection and decision tables, puzzles; Inductive reasoning: Assessment of ability to learn by example, imitation or by trial – Analogy pattern recognition, Classification pattern recognition, Coding pattern recognition, Number series pattern recognition; Abductive reasoning: Critical thinking ability of seeing through logical weak links or loopholes in an argument or a group of statements; Critical reasoning: assessment of ability to think through and analyze logical arguments, assessment of ability to use logical constructs to offer reasoning in unfamiliar situations; Information Gathering and synthesis: Ability of locating information, information ordering, rule based selection and data interpretation, order and classify data, interpret graphs, charts, tables and make rule based deductions. Application of these approaches for using visual, numerical and textual data from single or multiple sources

Unit 3: Quantitative Ability

Basic numbers – decimals and fractions, factorization, divisibility: HCF, LCM, Odd, even, prime and rational numbers. Application of algebra to real world, direct and inverse proportion, common applications – Speed-time -distance, Profit-loss, percentage, age relations, mixtures, other miscellaneous quantitative combination, exponentials and logarithms, permutations and combinations, probability. Spatial reasoning: Inductive – Missing portions, Sequence and series; Deductive analysis.

Reference Books –

18. "English Grammar in Use" by Raymond Murphy, Cambridge University Press.
19. "Word Power Made Easy" by Norman Lewis, Goyal Publishers & Distributors.
20. "Objective General English" by S.P. Bakshi, Arihant Publications.
21. "English for Competitive Examinations" by K. Sinha, S. Chand Publishing.
22. "Essential English Grammar" by Philip Gucker, Wiley.
23. "English Idioms and Phrasal Verbs" by M.A. Yadav, Vikas Publishing House.
24. "The Oxford English Grammar" by Sidney Greenbaum, Oxford University Press.
25. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121903409.

26. "Logical Reasoning and Data Interpretation for the CAT" by Nishit K. Sinha, Pearson India, ISBN: 978-8131709117.
27. "Logical Reasoning and Data Interpretation for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
28. "A New Approach to Reasoning Verbal and Non-Verbal" by B.S. Sijwali & Indu Sijwali, Arihant Publications, ISBN: 978-9311124692.
29. "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121900637.
30. "How to Prepare for Quantitative Aptitude for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
31. "The Pearson Guide to Quantitative Aptitude for Competitive Examination" by Pearson, Pearson India, ISBN: 978-8131709117.
32. "Quantitative Aptitude for Competitive Examinations" by Abhijit Guha, Tata McGraw Hill Education, ISBN: 978-0070666653.
33. "Data Interpretation & Data Sufficiency" by R.S. Aggarwal, S. Chand Publishing ISBN: 978-8121903515.
34. "Quantitative Aptitude for Competitive Examinations" by S. Chand, S. Chand Publishing, ISBN: 978-8121903423.

Course Outcomes:

Upon completion of the course, the student will be able to –

7. Improve the reading, writing and verbal skills, and enhance comprehension and articulation abilities
8. Develop logical reasoning abilities, enabling them to make sound decisions in problem-solving scenarios
9. Develop mathematical aptitude as well as data interpretation abilities and use them in test cases and real world problems
10. Learn to apply approaches for optimum time-management, prioritization maximizing the accuracy
11. Learn data interpretation, apply mathematical skills to draw accurate conclusions
12. Apply their knowledge of English, reasoning and quantitative skills for planning, critical thinking and real world problems

CO-PO Map

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2					3		2	2				3
CO2	2	2	3	2	2		2		3		2	2	3		3	3
CO3	2	2	3	2	3		2		3		2	2	3		3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3	3	3
CO5	2	2	3	2	3	2			3		2	2	3		3	3
CO6	2	2	3	3	2				3		3	2	3		3	3
Average	**2.0**	**2.0**	**2.83**	**2.83**	**2.6**	**2.5**	**2.0**	**2.0**	**3.0**	**1.0**	**2.16**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**

FF No. : 654

CB2001: Design Thinking - 2

Credits: 01 **Teaching Scheme: Tutorial 01 Hr/week**

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:
To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?
Importance of Paper Publication and Patents
Structure of Paper
Journal Publication
Publication in conference
Literature Review
Research Paper Writing
Journal Ratings and Evaluation
How to rate a Journal?
Intellectual property (IP)
Research Ethics
Entrepreneurship

Section 2: Topics/Contents

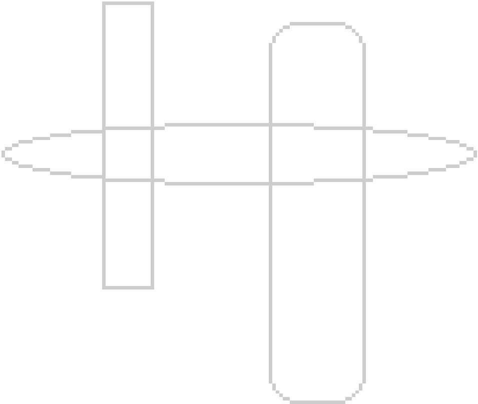
Structure of The paper
Journal List (Top 50 Journals)
Selection of the journal
Use of various online journal selection tools
Plagiarism checking
Improving contents of the paper
Patent drafting
Patent search
Filing of patent
Writing answers to reviewer questions
Modification in manuscript
Checking of publication draft

Course Outcome:[Publication of paper or patent]
The student will be able to
1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of itspublication
4. Write a Research Paper based on project work

5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**	
CO1	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	
CO2	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	
CO3	2	2	3	3	2	2	1	2	2	3	--	1	1	3	3	
CO4	3	3	3	3	3	2	1	2	2	3	1	1	1	3	3	
CO5	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	
CO6	2	2	2	2	2	2	1	3	2	3	--	1	1	3	3	
C07	1	1	1	1	1	--	--	--	--	--	--	1	1	1	1	
Average	**1.57**	**1.57**	**1.71**	**1.71**	**1.57**	**2.0**	**1.0**	**2.33**	**2.0**	**3.0**	**1.0**	**1.0**	**1.0**	**2.28**	**2.28**	



FF No. :654

CB2002: Engineering Design and Innovations- 2

Credits:.4 **Lab: 8 hours/Week**

Course Prerequisites: Problem Based Learning

Course Objectives:

1.To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2.To Evaluate alternative approaches, and justify the use of selected tools and methods,
3.To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4.To engage students in rich and authentic learning experiences.
5.To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of facultyand enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project centric learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Engineering Design" are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:
Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.
- It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL:

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution.
  How effective is …….? How strong is the evidence for ………? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- · How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDD Sample Case Studies : -

With the adaptation of industry communication standards, Raspberry Pi and Sensors, following projects can be taken up:

1) Design a deployable product for soil moisture detection
2) Design a deployable product for temperature detection
3) Design a deployable product for pressure detection
3) Design a deployable product smoke detection
4) Design a deployable product for motion detection
5) Design a deployable product for collision detection
6) Design a deployable product for sound detection

...not limited to.....Faculty and students are free to include other areas which meet the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

To focus on the higher levels of the Booms Taxonomy analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1.A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2.Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: ***(As per IEEE format)***

1.De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.

2.Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

MOOCs Links and additional reading material:www.nptelvideos.in
https://worldwide.espacenet.com/

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO POMap:

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	
CO1	2	2	2	2					3		2	2	2	3	3	
CO2	2	2	3	2	2		2		3		2	2	2	3	3	
CO3	2	2	3	2	3		2		3		2	2	2	3	3	
CO4	2	2	3	2	3	3		2	3		2	2	2	3	3	
CO5	2	2	3	2	3	2			3		2	2	2	3	3	
CO6	2	2	3	3	2				3		3	2	3	3	3	
Average	2.0	2.0	2.83	2.83	2.6	2.5	2.0	2.0	3.0	1.0	2.16	2.0	2.17	3.0	3.0	

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3