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 of Second Year CSE-IoTCSBT for Academic Year 2025-26

Course Code	Course Name	Type	Teaching Learning Scheme				
			Th	Tut	Lab	Hrs. / Week	Credits
CB2003	Fundamentals of Data Structures	PCC	3	0	2	5	4
CB2004	Data Communication and Networking	PCC	2	1	0	3	3
CB2005	Basics of IoT	PCC	2	0	2	4	3
CB2006	Object Oriented Programming	PCC	1	0	2	3	2
CBM001	Internet of Things	MDM	2	1	0	3	3
HS2002	From Campus to Corporate - 1	HSSM	2	0	0	2	2
HS2001	Reasoning and Aptitude Development - 3	VSEC	1	0	0	1	1
CB2001	Design Thinking – 3	AEC	0	1	0	0	1
CB2002	Engineering Design and Innovation - III	FP	0	0	4	4	2
							21

Assessment scheme

Course Code	Course Name	ESE TH (W)	ESE TH (O)	CVV	CP	LAB	GD/PPT/HA	MSE (O)	T1 (O)	T2 (O)	ESE	PRACT + CVV	Total
CB2003	Fundamentals of Data Structures	40		20	30	10							100
CB2004	Data Communication and Networking		40	30			GD/PPT 30						100
CB2005	Basics of IoT	40		20	30	10							100
CB2006	Object Oriented Programming				30	10						40 + 20	100
CBM001	Internet of Things						HA 30		35	35			100
HS2002	From Campus to Corporate - 1		50					50					100
HS2001	Reasoning and Aptitude Development - 3										100		100
CB2001	Design Thinking - 3										100		100
CB2002	Engineering Design and Innovation - III							30			70		100

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_0129720024067194883 7_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.

FF No. : 654

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	**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**

FF No. : 654

CB2001: Design Thinking - 3

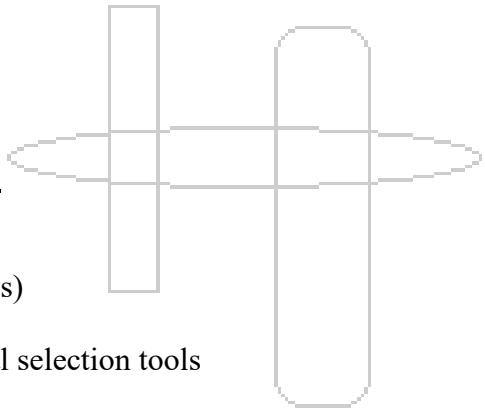
Credits: 1 **Teaching Scheme:** Tutorial: 1 Hour/Week

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 - III

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	**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**

CO attainment levels

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