Final

DIPLOMA CURRICULUM OF COMPUTER ENGINEERING AND IOT (SECOND YEAR) (3rd Semester)

(To be implemented from 2025-26)

Prepared by;

National Institute of Technical Teachers' Training & Research Kolkata
Block – FC, Sector – III, Salt Lake City, Kolkata – 700106

Vetted by:
Domain experts from Polytechnics of Odisha

State Council for Technical Education & Vocational Training
Near Raj Bhawan, Unit-VIII, Bhubaneswar, Odisha

Table of Contents

Sl. No.	Contents	Page No.
1	Curriculum Structure for Second year (Semester III)	3
2	Detailed Course Contents of Semester III	4 - 25

PROGRAMME TITLE: COMPUTER ENGINEERING AND IOT

SEMESTER – III

SL. No	Category of Course	Code No	Course Title	Study Scheme				Evaluation Scheme				Total Marks	Credits
				Pre-requisite	Contact Hours/ week			Theory		Practical			
					L	T	P	End Exam	Progressive Assessment	End Exam	Progressive Assessment		
1	Programme Core	CEIPC201 TH:1	Programming with C++		3	0	0	70	30	-	-	100	3
2		CEIPC203 TH:2	IoT Fundamentals		3	0	0	70	30	-	-	100	3
3		CEIPC205 TH:3	Data Structures and Algorithms		3	0	0	70	30	-	-	100	3
4		CEIPC207 TH:4	Digital Electronics and Computer Organization		3	0	0	70	30	-	-	100	3
5		CEIPC209 TH:5	Introduction to Sensor Technology		3	0	0	70	30	-	-	100	3
6		CEIPC211 PR:1	Programming with C++ Lab		0	0	4	-	-	15	35	50	2
7		CEIPC213 PR:2	IoT Fundamentals Lab		0	0	4	-	-	15	35	50	2
8		CEIPC215 PR:3	Data Structures and Algorithms Lab		0	0	4	-	-	15	35	50	2
9		CEIPC217 PR:4	Digital Electronics & CO Lab		0	0	4	-	-	15	35	50	2
10	Summer Internship	SI 201	Summer Internship – I*		0	0	0	-	-	15	35	50	2
TOTAL					**15**	**0**	**16**	**350**	**150**	**75**	**175**	**750**	**25**

***4-week internship after 2nd Semester**

SEMESTER - III COURSES

TH:1- PROGRAMMING WITH C++

L	T	P	Total Marks: 100	Course Code: CEIPC 201/TH1	
3	0	0			
Total Contact Hours				**Theory Assessment**	
Theory		: 45Hrs		End Term Exam	: 70
				Progressive Assessment	: 30
Pre-Requisite		**: Nil**			
Credit		**: 3**		Category of Course	: PC

RATIONALE:

Programming with C++ is a foundation course for any would-be IT professional. It gives exposure to the basic techniques of computer programming in current technological scenario. This course is most essential for any curriculum of Computer Science and Engineering.

LEARNING OUTCOMES:

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

- Describe object-oriented programming (OOP) principles.
- Develop proficiency in C++ syntax and programming constructs.
- Implement advanced OOP features for software design.
- Demonstrate polymorphism and operator overloading.
- Handle exceptions and ensure robust program execution.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	Introduction to object oriented programming, user defined types, structures, unions, polymorphism, encapsulation. Getting started with C++ syntax, data-type, variables, strings, functions, default values in functions, recursion, namespaces, operators, flow control, arrays and pointers.	12
II	Abstraction mechanism: Classes, private, public, constructors, destructors, member data, member functions, inline function, friend functions, static members, and references. Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors.	11
III	Polymorphism: Binding, Static binding, Dynamic binding, Static polymorphism: Function Overloading, Ambiguity in function overloading, Dynamic polymorphism: Base class pointer, object slicing, late binding, method overriding with virtual functions, pure virtual functions, abstract	9

	classes.	
IV	Operator Overloading: This pointer, applications of this pointer, Operator function, member and non-member operator function, operator overloading, I/O operators.	9
V	Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration, unexpected exceptions	4

REFERENCES:

1.	Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India)
2.	ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education
3.	C++: The Complete Reference - Schildt, McGraw-Hill Education (India)
4.	C++ and Object Oriented Programming - Jana, PHI Learning.
5.	Object Oriented Programming with C++ - Rajiv Sahay, Oxford
6.	Mastering C++ - Venugopal, McGraw-Hill Education (India)

TH:2- IOT FUNDAMENTALS

L	T	P	Total Marks: 100	Course Code: CEIPC 203/TH2	
3	0	0			
Total Contact Hours				**Theory Assessment**	
Theory		: 45 Hrs		End Term Exam	: 60
				Progressive Assessment	: 40
Pre-Requisite		**: Nil**			
Credit		**: 3**		Category of Course	: PC

RATIONALE:

This course will take care about the fundamentals of IoT technology, which is now widely used as one of the established technologies in modern life. Various electronics gadgets are now being activated through sensors and used through Internet remotely. This combination has caused spectacular changes in the fields of agriculture, home automation, energy sector, health services and many more. This course on IoT fundamentals provides an elementary concept of the functioning and application of this new internet and sensor-based technology.

LEARNING OUTCOMES:

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

- Describe the fundamental concepts, architecture, and protocols of the Internet of Things (IoT).
- Analyze IoT applications across various domains, including home automation, healthcare, and industry.
- Differentiate between Machine-to-Machine (M2M) communication and IoT, and explore SDN and NFV for IoT.
- Explain system management techniques using SNMP, NETCONF, and YANG for efficient IoT deployment.
- Apply IoT development methodologies, including requirement analysis, domain modeling, and service specification.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to IoT:** Definition and characteristics of IoT, Functional requirements and motivation for IoT, Physical design: Things in IoT and their roles, Basic concepts of internetworking and IoT architecture, Overview of TCP/IP and IoT-specific protocols	8
II	**IoT logical design and communication:** Functional blocks of IoT architecture, Communication models and APIs for IoT, IoT protocols and their comparison, Overview of different networking technologies in IoT	8
III	**IoT applications**: Home Automation: Smart lighting, security, and appliances, Smart Cities: Traffic management, smart parking, waste	7

	management, Environmental Monitoring: Pollution control, weather tracking, Energy Sector: Smart grids and energy optimization, Healthcare: Remote monitoring and wearable devices, Industrial IoT: Predictive maintenance, automation	
IV	**M2M communication and IoT system management**: Introduction to Machine-to-Machine (M2M) communication, Difference between M2M and IoT, Software-Defined Networking (SDN) and its role in IoT, Network Function Virtualization (NFV) in IoT, IoT system management: Need and importance, Management protocols: SNMP, NETCONF, YANG	8
V	**IoT development methodology**: IoT development lifecycle, Requirements and constraints for IoT applications, Process and domain model specifications, Information model specification, Service specification for IoT	7
VI	**IoT architecture and security:** IoT level specifications and their importance, IoT security challenges and solutions, Secure data transmission in IoT, Privacy concerns in IoT applications, Future trends and emerging technologies in IoT	7

REFERENCES:

1.	Dr. Ovidiu Vermesan and Dr. Peter Friess, Internet of Things: From research and innovation to market deployment, River Publishers 2014.
2.	Honbo Zhou,"The Internet of Things in the Cloud: A Middleware Perspective" - CRC Press 2012.
3.	Arshdeep Bahga and Vijay Madisetti, Internet of Things A Hand-on Approach, Universities press, 2015
4.	Pethuru Raj and Anupama C.Raman, "The Internet of Things: Enabling Technologies and Use Cases, CRC Press
5.	Dieter Uckelmann et.al, Architecting the Internet of Things, Springer, 2011

TH:3- DATA STRUCTURES AND ALGORITHMS

L	T	P	Total Marks: 100	Course Code: CEIPC 205/TH3	
3	0	0			
Total Contact Hours				**Theory Assessment**	
Theory		: 45Hrs		End Term Exam	: 70
				Progressive Assessment	: 30
Pre-Requisite		**: Nil**			
Credit		**: 3**		Category of Course	: PC

RATIONALE:

Data Structures and Algorithms (DSA) form the backbone of computer science and software engineering. They enable efficient data storage, retrieval, and manipulation, while algorithms provide systematic methods to solve complex computational problems. Mastering DSA equips students with problem-solving skills essential for developing optimized, scalable, and robust software systems.

LEARNING OUTCOMES:

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

- Define the basic terminologies and classifications of data structures.
- Explain the representation and operations of linear data structures like stacks, queues, and linked lists.
- Implement algorithms for sorting and searching using appropriate data structures.
- Analyze the time and space complexity of data structure operations and algorithm paradigms like dynamic programming and backtracking.
- Design solutions using advanced data structures for real-world applications, such as shortest path problems or flow-based algorithms.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Data Structures:** Basic Terminology, Classification of Data Structure, Operations on Data Structure, Asymptotic and worst-case analysis of algorithms.	6
II	**Linear Data Structures:** Stacks-Introduction to Stacks, Array Representation of Stacks, Operations on a Stack, Applications of Stacks-Infix-to-Postfix Transformation, evaluating Postfix Expressions. Queues: Introduction to Queues, Array Representation of Queues, Operations on a Queue, Types of Queues-DeQueue, Circular Queue, Applications of Queues-Round Robin Algorithm.	7
III	**Linked Lists:** Singly Linked List, Representation in Memory, Operations on a Single Linked List, Circular Linked Lists, Doubly Linked Lists, Linked List Representation and Operations of Stack, Linked List	6

	Representation and Operations of Queue.	
IV	**Non-Linear Data Structures:** Trees-Basic Terminologies, Definition and Concepts of Binary Trees, Representations of a Binary Tree using Arrays and Linked Lists, Operations on a Binary Tree-Insertion, Deletion, Traversals, Types of Binary Trees. GRAPHS: Graph Terminologies, Representation of Graphs- Set, Linked, Matrix, Graph Traversals	7
V	**Algorithm Paradigms:** Greedy, Divide and Conquer, Branch and Bound, Dynamic Programming and Backtracking.	5
VI	**Sorting:** The sorting problem. Bubble sort, Selection sort, Insertion sort, Mergesort, Quicksort, Searching: Symbol Tables, Binary Search Trees, Balanced Search Trees. Hash Tables.	8
VII	**Graphs:** Definition of a directed and undirected graph. Paths, Cycles, spanning trees, Directed Acyclic Graphs. Topological Sorting. Minimum Spanning Tree algorithms, Shortest Path algorithms: Dijkstra's algorithm. Flow-based algorithms.	6

REFERENCES:

1.	Narasimha Karumanchi, Data Structures And Algorithms Made Easy: Data Structures And Algorithmic Puzzles, 2nd Edition, CareerMonk Publications, India, 2011.
2.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 3rd Edition, MIT Press, Cambridge, 2009.
3.	Reema Thareja, Data Structures Using C, 2nd Edition, Oxford University Press India, New Delhi, 2014.
4.	Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, Fundamentals of Data Structure in C, 2nd Edition, University Press, India, 2008.
5.	Gajendra Sharma, Design & Analysis of Algorithms, 1st Edition, Khanna Publishing House, New Delhi, 2016.
6.	Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Pearson Education, United States, 2011.

TH:4- DIGITAL ELECTRONICS AND COMPUTER ORGANISATION

L	T	P		Course Code: CEIPC 207/TH4	
3	0	0			
Total Contact Hours				**Theory Assessment**	
Theory		: 45Hrs	**Total Marks: 100**	End Term Exam	: 70
				Progressive Assessment	: 30
Pre Requisite		**: Nil**			
Credit		**: 3**		Category of Course	: PC

RATIONALE:

Digital Electronics and Computer Organization form the foundation of modern computing, enabling the design and functioning of digital systems. They cover essential concepts like logic circuits, memory, and system architecture, bridging hardware and software integration. Learning these topics equips students with the skills to understand and develop efficient, reliable computing systems.

LEARNING OUTCOMES:

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

- Define key concepts of digital electronics, including number systems, Boolean algebra, and logic gates.
- Explain the principles behind combinational and sequential circuits, such as multiplexers, flip-flops, and counters, and their applications.
- Implement simplified logic circuits using Karnaugh Maps and Boolean algebra to solve real-world digital design problems.
- Analyze the instruction cycle, memory organization, and processor architecture to evaluate system performance and identify bottlenecks.
- Design to simulate a basic CPU operation or create a functional digital circuit using the concepts of digital electronics and computer organization.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Digital Electronics:** Difference Between Analog and Digital Signals Number Systems: Binary, Octal, Decimal, and Hexadecimal Conversion Between Number Systems, Binary Arithmetic Boolean Algebra: Basic Operations, Laws, and Simplification	8
II	**Logic Gates and Circuits: Logic Gates:** AND, OR, NOT, NAND, NOR, XOR, XNOR, Design and Simplification of Logic Circuits Using Boolean Algebra, Karnaugh Maps (K-Maps) for Simplification Practical Applications of Logic Gates in Real-World Circuits	7

III	**Combinational and Sequential Circuits:** Combinational Circuits: Multiplexers, Demultiplexers, Encoders, and Decoders, Sequential Circuits: Flip-Flops (SR, JK, D, T) and Their Applications Counters: Synchronous and Asynchronous Counters Registers and Shift Registers: Types and Uses	8
IV	**Fundamentals of Computer Organization:** Basic Structure of a Computer: CPU, Memory, Input/Output Devices Instruction Cycle: Fetch, Decode, Execute Memory Organization: Types of Memory (RAM, ROM, Cache, Virtual Memory), Introduction to Buses: Address Bus, Data Bus, and Control Bus	8
V	**Processor Architecture and Control:** Unit Introduction to Microprocessors and Microcontrollers, Basics of Arithmetic Logic Unit (ALU) and Control Unit, Instruction Set Architecture (ISA): RISC vs CISC Pipelining and Performance Optimization in Processors	7
VI	**Input/Output Systems and Advanced Topics:** I/O Devices and Interfaces: Keyboard, Mouse, Printers, and Storage Devices Interrupts and DMA (Direct Memory Access), Overview of Modern Trends: Multicore Processors, GPUs, and Embedded Systems, Mini- Project: Design a Simple Digital Circuit or Simulate a Basic CPU Operation	7

REFERENCES:

1.	M. Morris Mano, Digital Design, 5th Edition, Pearson Education, India, 2013.
2.	David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th Edition, Morgan Kaufmann, San Francisco, 2013.
3.	V. Rajaraman and T. Radhakrishnan**,** Digital Logic and Computer Organization, 1st Edition, PHI Learning, New Delhi, 2006.
4.	R. P. Jain**,** Modern Digital Electronics, 4th Edition, Tata McGraw-Hill Education, New Delhi, 2009.
5.	Dhananjay M. Dhamdhere**,** Computer Organization and Assembly Language Programming, 1st Edition, Tata McGraw-Hill Education, New Delhi, 1987.

TH:5- INTRODUCTION TO SENSOR TECHNOLOGY

L	T	P	Total Marks: 100	Course Code: CEIPC 209/TH5	
3	0	0			
Total Contact Hours				**Theory Assessment**	
Theory		: 45 Hrs		End Term Exam	: 70
				Progressive Assessment	: 30
Pre-Requisite		**: Nil**			
Credit		**: 3**		Category of Course	: PC

RATIONALE:

Use of sensor along with embedded system can be considered as the building block of IoT. The entire gamut of IoT technology is pivoted on the effective use of sensors put into embedded system. Adequate exposure in sensor technology is a pre-requisite for understanding appropriate application of IoT in different fields like agriculture, health care, energy sector etc.

LEARNING OUTCOMES:

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

- Describe the concepts of Embedded Systems.
- Discuss the Off-The-Shelf Components and interface.
- Interpret the design of RTOS.
- Illustrate the different types of Sensors and Transducers.
- Infer the data accuracy, real-time monitoring, and connectivity using smart sensors.
- Apply autonomous functionality across diverse applications.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**INTRODUCTION TO EMBEDDED SYSTEMS** - Definition of Embedded System - Embedded Systems Vs General Computing Systems - History of Embedded Systems - Classification - Major Application Areas - Purpose of Embedded Systems - Characteristics and Quality Attributes of Embedded Systems	6
II	**TYPICAL EMBEDDED SYSTEM** - Core of the Embedded System: General Purpose and Domain Specific Processors, ASICs, PLDs, Commercial Off-The-Shelf Components (COTS) - Memory: ROM, RAM - Memory according to the type of Interface - Memory Shadowing	6

	- Memory selection for Embedded Systems - Sensors and Actuators - Communication Interface: On board and External Communication Interfaces	
III	**RTOS BASED EMBEDDED SYSTEM DESIGN** - Operating System Basics - Types of Operating Systems - Tasks, Process and Threads - Multiprocessing and Multitasking - Task Scheduling	3
IV	**SENSORS / TRANSDUCERS** - Principles - Classification - Parameters - Characteristics - Environmental Parameters (EP) - Characterization Mechanical and Electromechanical Sensors: Introduction - Resistive Potentiometer - Strain Gauge - Resistance Strain Gauge - Semiconductor Strain Gauges - Inductive Sensors: Sensitivity and Linearity of the Sensor - Types-Capacitive Sensors: Electrostatic Transducer - Force/Stress Sensors Using Quartz Resonators - Ultrasonic Sensors - Thermal Sensors: Introduction - Gas Thermometric Sensors - Thermal Expansion Type Thermometric Sensors - Acoustic Temperature Sensor - Junction Semiconductor Types - Thermal Radiation Sensors - Quartz Crystal Thermoelectric Sensors - NQR Thermometry - Spectroscopic Thermometry - Noise Thermometry - Heat Flux Sensors - Magnetic sensors **Introduction -** Sensors and the Principles Behind** - Magneto resistive Sensors - Anisotropic Magneto resistive Sensing - Semiconductor Magneto resistors - Hall Effect and Sensors - Inductance and Eddy Current Sensors - Angular/Rotary Movement Transducers - Synchros - Synchro-resolvers - Eddy Current Sensors	15

	- Electromagnetic Flowmeter - Switching Magnetic Sensors - SQUID Sensors	
V	**RADIATION SENSORS AND SMART SENSORS** - Introduction - Basic Characteristics - Types of Photo sensors/Photo detectors - X-ray and Nuclear Radiation Sensors - Fiber Optic Sensors Electro analytical Sensors: Introduction - The Electrochemical Cell - The Cell Potential - Standard Hydrogen Electrode (SHE) - Liquid Junction and Other Potentials - Polarization - Concentration Polarization - Reference Electrodes - Sensor Electrodes - Electro ceramics in Gas Media. Introduction - Primary Sensors - Excitation - Amplification - Filters - Converters - Compensation - Information Coding/Processing - Data Communication - Standards for Smart Sensor Interface - The Automation Sensors - Applications: Introduction - On-board Automobile Sensors (Automotive Sensors) - Home Appliance Sensors - Aerospace Sensors - Sensors for Manufacturing - Sensors for environmental Monitoring	15

REFERENCES:

1.	Introduction to Embedded Systems - Shibu K.V, McGraw Hill.
2.	Sensors and Transducers - D. Patranabis PHI Learning Private Limited.
3.	Embedded Systems - Raj Kamal, TMH.

PR:1- PROGRAMMING WITH C++ Lab

L	T	P		Course Code: CEIPC 211/PR1	
0	0	4			
Total Contact Hours				**Practical Assessment**	
Practical		: 60Hrs	**Total Marks: 50**	End Term Exam	: 15
				Progressive Assessment	: 35
Pre-Requisite		**: Nil**			
Credit		**: 2**		Category of Course	: PC

RATIONALE:

Programming with C++ is a foundation course for any would-be IT professional. It gives exposure to the basic techniques of computer programming in current technological scenario. It is most essential for any curriculum of Computer Science and Engineering. This course provides the student with the skill to write computer programming with C++ language.

Learning Objectives:

After completing this course, students will be able to:

- Implement object-oriented programming (OOP) concepts like encapsulation, inheritance, and polymorphism.
- Use control structures, functions, and recursion to solve programming problems efficiently.
- Work with C++ Standard Library components, including file handling and STL.
- Apply problem-solving techniques through hands-on coding exercises and small projects.
- Develop skills in writing, compiling, debugging, and executing C++ programs.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	Familiarization with C++ programming environment (Editor, Compiler, etc.)	4
II	Programs using, I/O statements and various operators in C++	4
III	Programs using C++ expression evaluation and precedence	4
IV	Programs using C++ decision-making statements and branching statements	4
V	Programs using loop statements in C++	4
VI	Programs to demonstrate applications of n-dimensional arrays in C++	4
VII	Programs to demonstrate use of string manipulation functions in C++	4
VIII	Programs to demonstrate parameter passing mechanism in C++	4

IX	Programs in C++ to demonstrate recursion	4
X	Programs in C++ to demonstrate use of pointers	4
XI	Programs in C++ to demonstrate class, object, constructor, destructor.	8
XII	Programs in C++ to demonstrate function overloading, operator overloading.	6
XIII	Programs in C++ to demonstrate dynamic memory allocation, file operations	6

REFERENCES:

1.	Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education
2.	ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education
3.	C++: The Complete Reference - Schildt, McGraw-Hill Education (India)
4.	C++ and Object Oriented Programming - Jana, PHI Learning.
5.	Object Oriented Programming with C++ - Rajiv Sahay, Oxford
6.	Mastering C++ - Venugopal, McGraw-Hill Education (India)

PR:2- IOT FUNDAMENTALS LAB

L	T	P		Course Code: CEIPC 213/PR2	
0	0	4			
Total Contact Hours			**Total Marks: 50**	**Practical Assessment**	
Practical		: 60Hrs		End Term Exam	: 15
				Progressive Assessment	: 35
Pre-Requisite		**: Nil**			
Credit		**: 2**		Category of Course	: PC

RATIONALE:

The IoT Fundamentals Lab provides hands-on experience in designing, configuring, and implementing IoT systems. It helps students understand the integration of sensors, actuators, and communication protocols essential for IoT applications. This lab equips students with the foundational skills needed to develop and troubleshoot IoT solutions in real-world scenarios.

LEARNING OUTCOMES:

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

- Describe the fundamentals of IoT, its architecture, and its integration with multimedia technologies.
- Explain various communication protocols and wireless technologies used in IoT for transmitting multimedia data.
- Identify different types of multimedia data (audio, video, images) and the associated sensors and techniques for their acquisition and processing in IoT systems.
- Evaluate the benefits and challenges of edge computing for processing multimedia data in IoT applications.
- Analyze the role of cloud platforms in managing, storing, and streaming multimedia content in IoT ecosystems, with an emphasis on latency optimization.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	Controlling the Light Emitting Diode (LED) with a push button.	4
II	Interfacing the RGB LED with the Arduino	8
III	Controlling the LED blink rate with the potentiometer interfacing with Arduino	8
IV	Detection of the light using photo resistor	8
V	Interfacing of temperature sensor LM35 with Arduino	8
VI	Interfacing Servo Motor with the Arduino	8
VII	Interfacing of the Active Buzzer with Arduino.	8
VIII	Interfacing of the Relay with Arduino.	8

REFERENCES:

1.	"Internet of Things: A Hands-On Approach" by Arshdeep Bahga and Vijay Madisetti
2.	"Practical Internet of Things" by Shancang Li and Li Da Xu
3.	"Internet of Things: Architecture and Design Principles" by Rajkumar Buyya, Amir Vahid Dastjerdi
4.	"IoT Projects with Arduino, Raspberry Pi, and ESP8266" by Agus Kurniawan
5.	"Raspberry Pi IoT Projects" by John C. Shovic and Aidan Finn

PR:3- DATA STRUCTURES AND ALGORITHMS LAB

L	T	P		Course Code: CEIPC 215/PR3	
0	0	4			
Total Contact Hours				**Practical Assessment**	
Practical		: 60Hrs	**Total Marks: 50**	End Term Exam	: 15
				Progressive Assessment	: 35
Pre-Requisite		**: Nil**			
Credit		**: 2**		Category of Course	: PC

RATIONALE:

Data Structures and Algorithms (DSA) form the backbone of computer science and software engineering. They enable efficient data storage, retrieval, and manipulation, while algorithms provide systematic methods to solve complex computational problems. Mastering DSA equips students with problem-solving skills essential for developing optimized, scalable, and robust software systems.

LEARNING OUTCOMES:

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

- Explain basic terminologies and operations on data structures.
- Perform asymptotic and worst-case analysis of algorithms.
- Implement linear data structures.
- Apply trees and graphs to solve problems.
- Implement sorting and searching algorithms.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Introduction to Data Structures:** • Write a program to analyze and compare the time complexity of basic operations (e.g., searching, insertion) on arrays and linked lists.	2
II	**Linear Data Structures:** • Implement stack operations (push, pop, peek) using arrays and linked lists • Develop programs for applications of stacks (e.g., infix-to-postfix conversion and postfix evaluation) • Implement queue operations (enqueue, dequeue) using arrays and linked lists • Write programs for types of queues: circular queues and dequeue	12

III	**Linked Lists:** • Implement singly linked list operations (insertion, deletion, traversal). • Write programs to create and manipulate circular and doubly linked lists • Implement stack and queue operations using linked lists.	10
IV	**Non-Linear Data Structures:** • Implement binary tree operations (insertion, deletion, traversal) • Develop programs for types of binary trees (binary search tree, AVL tree) • Implement graph representations (adjacency list, adjacency matrix) • Basic graph traversals (BFS, DFS).	16
V	**Sorting:** • Implement sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort.	10
VI	**Searching:** • Write programs for searching using binary search trees (BST) and hash tables • Implement symbol table operations using balanced search trees.	10

REFERENCES:

1.	Seymour Lipschutz , Schaum's Outlines - Data Structures With C, Tata-MacGraw-Hill
2.	Sahni Sartaj, Data Structures, Algorithms and Applications in C++, 2nd Edition, Universities Press, India, 2005.
3.	Yashavant Kanetkar, Data Structures Through C, 2nd Edition, BPB Publications, India, 2008.
4.	Reema Thareja, Data Structures Using C, 2nd Edition, Oxford University Press, India, 2014.
5.	Debasis Samanta, Classic Data Structures, 2nd Edition, PHI Learning Pvt. Ltd., India, 2009.
6.	A. M. Tenenbaum, Data Structures Using C and C++, 2nd Edition, Pearson Education, India, 2003.
7.	Sahni Sartaj, Data Structures, Algorithms and Applications in C++, 2nd Edition, Universities Press, India, 2005.

PR:4- DIGITAL ELECTRONICS & CO LAB

L	T	P		Course Code: CEIPC 217/PR4	
0	0	4			
Total Contact Hours				**Practical Assessment**	
Practical		: 60Hrs	**Total Marks: 50**	End Term Exam	: 20
				Progressive Assessment	: 30
Pre-Requisite		**: Nil**			
Credit		**: 2**		Category of Course	: PC

RATIONALE:

This lab helps students gain practical skills in logic gates, combinational and sequential circuits, flip-flops, counters, and memory devices. It provides hands-on experience with digital ICs and programmable logic devices, building a strong foundation for embedded systems, automation, and digital system design.

LEARNING OUTCOMES:

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

- Explain basic logic gates and Boolean algebra principles.
- Design combinational circuits like adders, multiplexers, and decoders.
- Analyze and implement sequential circuits such as flip-flops, counters, and shift registers.
- Work with timing circuits using the 555 timer and clock pulse generators.
- Interface memory and programmable logic devices (PLDs) in digital circuits.
- Apply digital electronics concepts to real-world applications like ALUs and traffic light controllers.

DETAILED COURSE CONTENTS:

Unit No.	Topic/Sub-Topic	Allotted Time (Hours)
I	**Number Systems and Logic Gates** • Number Systems & Conversions (Binary, Decimal, Octal, Hexadecimal) • Boolean Algebra & Simplification • Logic Gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) • Universal Gates (NAND & NOR) Implementation	10
II	**Combinational Logic Circuits** • Half Adder and Full Adder • Half Subtractor and Full Subtractor • 4-bit Parallel Adder/Subtractor (Using IC 7483) • Multiplexers (MUX) and Demultiplexers (DEMUX)	10

	• Encoders and Decoders • 7-Segment Display Decoder and Implementation	
III	**Sequential Logic Circuits** • Flip-Flops (SR, JK, D, T) – Truth Table & Applications • Master-Slave Flip-Flops • Asynchronous and Synchronous Counters • Up/Down Counters (Binary & Decade) • Shift Registers (SISO, SIPO, PISO, PIPO)	10
IV	**Timing and Clock Circuits** • 555 Timer as Monostable and Astable Multivibrator • Clock Pulse Generation using IC 555 • Frequency Division using Flip-Flops	10
V	**Memory and Programmable Logic Devices** • Basic Concept of RAM, ROM, EEPROM • Interfacing Memory with Digital Circuits • Introduction to PLD (PLA, PAL, FPGA)	10
VI	**Digital Logic Applications** • ALU Design Basics • Digital Comparator (2-bit & 4-bit) • Parity Generator and Checker • Traffic Light Control using Logic Gates	10

REFERENCES:

1.	Digital Electronics: Principles and Applications – A.P. Malvino, McGraw Hill Education, India, 2019
2.	Modern Digital Electronics – R.P. Jain, McGraw Hill Education, India, 2020
3.	Digital Logic and Computer Design – M. Morris Mano, Pearson Education, India, 2017
4.	Fundamentals of Digital Circuits – A. Anand Kumar, PHI Learning, India, 2022
5.	Digital Electronics – S. Salivahanan & S. Arivazhagan, Vikas Publishing, India, 2018

SUMMER INTERNSHIP - I

L	T	P		Course Code: SI 201	
0	0	0			
Total Contact Hours			**Total Marks: 50**		
Practical		: 30Hrs		End Term Exam	: 15
				Progressive Assessment	: 35
Pre-Requisite		**: Nil**			
Credit		**: 2**		Category of Course	: SI

Duration: 3-4 weeks during summer vacation after 2nd Semester.

RATIONALE:

Summer Internship - I is to offer a structured and practical learning experience that prepares individuals for their future careers, helps them make informed career choices, and equips them with the skills and knowledge necessary to succeed in their chosen field. This course provides opportunities to students for hands-on industry experience.

LEARNING OUTCOMES:

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

- Apply theoretical knowledge gained in their academic coursework to real-world
- situations.
- Develop and refine specific skills relevant to the field.
- Gains hands-on experience in a professional network by interacting with mentors
- and industry professionals.
- Learn to manage their time effectively.
- Clarify career goals.

DETAILED COURSE CONTENTS

SUGGESTED ACTIVITIES:

I Orientation:

- Introduction to the organization's mission, values, and culture.
- Familiarization with workplace policies, procedures, and safety guidelines.
- Orientation to the team and organizational structure.

II Project-Based Learning:

- Description of the main project or tasks the intern will be working on during the internship.
- Detailed project goals and objectives.
- Training and guidance on project-specific tools, technologies, or
- methodologies.

III Technical and Skill Development:

- Training sessions or workshops to enhance technical skills relevant to the internship role

(e.g., programming languages, software tools, laboratory techniques). • Soft skills development, including communication, teamwork, problem solving, and time management
IV Mentorship and Supervision: • Regular meetings with a designated mentor or supervisor for guidance, feedback, and support. • Mentorship objectives and expectations.
V Professional Development: • Sessions on professional etiquette, networking, and building a personal brand • Resume writing and interview preparation workshops.
VI Industry and Field-Specific Knowledge: • Lectures, seminars, or presentations on industry trends, best practices, and emerging technologies. • Guest speakers from the field to share insights and experiences.
VII Reporting and Documentation: • Training on how to document project progress, results, and findings. • Practice in creating reports, presentations, or other deliverables.
VIII Ethics and Professionalism: • Discussions on ethical considerations within the field. • Scenarios and case studies related to ethical decision-making
IX Feedback and Evaluation: • Regular performance evaluations and feedback sessions. • Self-assessment and goal-setting exercises.
X Networking and Industry Exposure: • Opportunities to attend industry conferences, webinars, or networking events. • Encouragement to connect with professionals in the field.

NOTE

As per AICTE guidelines, in Summer Internship-I, students are required to be involved in Inter/ Intra Institutional Activities viz;

- Training with higher Institutions;
- Soft skill training organized by Training and Placement Cell of the respective institutions;
- contribution at incubation/ innovation /entrepreneurship cell of the institute;
- participation in conferences/ workshops/ competitions etc.;
- Learning at Departmental Lab/ Tinkering Lab/ Institutional workshop;
- Working for consultancy/ research project within the institutes and
- Participation in all the activities of Institute's Innovation Council for eg: IPR workshop/Leadership Talks/ Idea/ Design/ Innovation/ Business Completion/ Technical Expos etc.