गोंय विद्यापीठ
ताळगांव पठार,
गोंय – ४०३ २०६
फोन : +९१-८६६९६०९०४८

International Year of Cooperatives
Cooperatives Build a Better World

(Accredited by NAAC)

Goa University
Taleigao Plateau, Goa-403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website : www.unigoa.ac.in

GU/Acad –PG/BoS -NEP/2025-26/274 Date: 30.07.2025

CIRCULAR

In continuation to the Circular No. GU/Acad –PG/BoS -NEP/2024/643 dated: 11.11.2024, the syllabus for Semester III & IV of the **Bachelor of Engineering in Information Technology** Programme approved by the Standing Committee of the Academic Council in its meeting held on 24th and 25th June 2025 is attached.

The Dean, Faculty of Engineering and Principals of affiliated Colleges offering the **Bachelor of Engineering in Information Technology** Programme are requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

ASHWIN VYAS LAWANDE
Digitally signed by ASHWIN VYAS LAWANDE
Date: 2025.07.30 11:27:03 +05'30'

(Ashwin V. Lawande)
Deputy Registrar – Academic

To,

1. The Dean, Faculty of Engineering, Goa University.
2. The Principals of affiliated Engineering Colleges.

Copy to,

1. The Director, Directorate of Technical Education, Govt. of Goa
2. The Chairperson, BoS in Information Technology Engineering.
3. The Controller of Examinations, Goa University.
4. The Assistant Registrar, Prof. Examinations (Technical and Allied), Goa University.
5. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

INFORMATION TECHNOLOGY SCHEME AY 2024-25

SEMESTER - III							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**Cre**
1	**Major**	**ITH-200**	Fundamentals of Data Structures using C++	3	0	0	3
		ITH-201	Fundamentals of Data Structures using C++ Lab	0	0	1	1
		ECM-202	Logic Design	3	0	0	3
		ECM-203	Logic Design Lab	0	0	1	1
2	**Minor/IC/ PE**	**ITH-221**	Computer Organisation and Architecture	3	0	0	3
		ITH-222	Computer Organisation and Architecture Lab	0	0	1	1
		OR					
		ITH-223	Embedded System Design and Applications	3	0	0	3
		ITH-224	Embedded System Design and Applications Lab	0	0	1	1
3	**Multi-disciplinary**	**SHM-234**	Engineering Mathematics II	3	0	0	3
4	**AEC**	**AEC-251**	*	0	0	2	2
5	**SEC**	**SEC-241**	App Development	0	0	3	3
			TOTAL	**12**	**0**	**8**	**20**

SEMESTER - IV							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**Credits**
1	**Major**	**ITH-204**	Object-Oriented Programming using JAVA	2	0	0	2
		ITH-205	Object-Oriented Programming using JAVA Lab	0	0	2	2
		ITH-206	Algorithm Design and Analysis	3	0	0	3
		ITH-207	Algorithm Design and Analysis Lab	0	0	1	1
		ITH-208	Fundamentals of Operating Systems	3	0	0	3
		ITH-209	Fundamentals of Operating Systems Lab	0	0	1	1
		ECM-208	Engineering Mathematics III	3	1	0	4
2	**Professional Electives**	**ITH-225**	Principles of Computer Graphics	3	0	0	3
		ITH-226	Principles of Computer Graphics Lab	0	0	1	1
		OR					
		ITH-227	Computer Networks	3	0	0	3
		ITH-228	Computer Networks Lab	0	0	1	1
			TOTAL	**14**	**1**	**5**	**20**

SEMESTER III

Major Courses

Name of the Programme : B.E in Information Technology
Course Code : ITH-200
Title of the Course : Fundamentals of Data Structures using C++
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	Knowledge of C programming	
Course Objectives:	This course enables students to: 1. Understand fundamental data structures and their roles in computation. 2. Describe linear and non-linear data structures with real-world relevance. 3. Analyze various sorting, searching, and hashing algorithms. 4. Apply suitable data structures to solve computational problems.	
Content:		**No of Hours**
Unit - 1	**Introduction To Data Representation & Data Structures:** Representation of arrays and their applications. **Stacks:** representation of stacks and its applications, Recursion, Tower of Hanoi, Implementation of recursive procedures by stacks. **Queues:** representation of queues and its applications, circular queues, priority queues, dequeue.	**12**
Unit - 2	**Lists:** Singly linked list, doubly linked list, circular linked list, linked stacks and queues and its applications. **Trees:** Basic terminology, binary trees and their representations, traversals of trees, applications of trees – infix/postfix representation if expressions and interconversion, B-tree, AVL.	**11**
Unit - 3	**Sorting:** Basic concept, Exchange sort, Selection sort, Insertion sort, Quick sort, Tree sort, Merge sort, Radix sort, Heaps and Heap sort. **Searching:** Basic searching techniques, sequential and binary search, tree searching. **Hashing:** Hash function, collision handling mechanisms.	**11**
Unit - 4	**Graphs:** Basic terminology, representation of graphs, directed and undirected graphs and their traversals, depth first and breadth first search, spanning trees. **Applications of Graphs:** Shortest path problem, topological sorting, matching.	**11**

Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.
References/ Readings	**Text books** 1. S. K. Srivastava; Data Structures Through C In Depth, 1st Edition, BPB publications, 2011, ISBN-10: 8183330785. 2. Yedidyah Langson, Moshej Augenstein, Aaron M. Tenenbaum, Data Structures using C & C++, 2nd Edition, Prentice Hall of India, 2000, ISBN-10: 8120315737. **Reference books** 1. Ellis Hurwitz and Sartaj Sahni; Fundamentals of Data Structures, 1st Edition, Golgotha Publications, 1984. 2. Sahni; Data Structures, Algorithms and Applications in C++, 2nd Edition, MGH, 2000, ISBN-10: 007049539X. 3. Yashawant Kanetkar; Data Structures Through C++, 5th Edition, BPB Publication, 2023, ISBN-10: 938928623X.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Identify basic data structures and their representations. CO 2. Explain the operations and applications of trees and graphs. CO 3. Analyze the efficiency of sorting and searching algorithms. CO 4. Apply appropriate data structures to solve defined computing problems.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-201**
Title of the Course : **Fundamentals of Data Structures using C++ Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Knowledge of C programming	
Course Objectives:	This course will enable students to: 1. Understand the implementation of linear and non-linear data structures. 2. Apply appropriate algorithms for searching, sorting, and traversal operations. 3. Analyze the efficiency of different data structure operations. 4. Implement dynamic memory techniques using linked data structures.	
Contents:	**List of Programs**	**No of Hours**
	1. Implementation of Stack using Arrays. 2. Implementation of queue using Linked lists. 3. Implementation of Binary Search tree & its operations & Traversals. 4. Implementation of Graph traversal techniques. 5. Implementation of Sorting techniques: Insertion Sort and Heap Sort. 6. Implementation of Sorting techniques: Merge Sort and Quick Sort. 7. Implementation of queue using Arrays. 8. Implementation of Doubly Linked Lists. 9. Implementation of Circular Queues. 10. Implementation of Search techniques: Linear Search and Binary Search.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**Text books** 1. S. K. Srivastava; Data Structures Through C In Depth, 1st Edition, BPB publications, 2011, ISBN-10: 8183330785. 2. Yedidyah Langson, Moshej Augenstein, Aaron M. Tenenbaum, Data Structures using C & C++, 2nd Edition, Prentice Hall of India, 2000, ISBN-10: 8120315737. **Reference books** 1. Ellis Hurwitz and Sartaj Sahni; Fundamentals of Data Structures, 1st Edition, Golgotha Publications, 1984. 2. Sahni; Data Structures, Algorithms and Applications in C++, 2nd Edition, MGH, 2000, ISBN-10: 007049539X.	

	3. Yashawant Kanetkar; Data Structures Through C++, 5th Edition, BPB Publication, 2023, ISBN-10: 938928623X.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Explain linear and non-linear data structures. CO 2. Analyze sorting and searching algorithm efficiency. CO 3. Apply suitable data structures to solve problems. CO 4. Implement programs using trees and graphs.

Name of the Programme : **B.E in Information Technology**
Course Code : **ECM-202**
Title of the Course : **Logic Design**
Number of Credits : **03**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basics of digital systems	
Course Objectives:	This course will enable students to: 1. Master number system conversions and complement arithmetic in digital Systems. 2. Simplify Boolean expressions and design combinational circuits using logic Gates. 3. Understand flip-flops and registers for data storage and sequence control. 4. Design and analyse sequential circuits, including state machines and Counters.	
Contents:		**No of Hours**
UNIT- 1	**Introduction**: Digital and Analog Systems, Logic Levels and Pulse Waveforms. **Number Systems:** Decimal, Binary, Octal, Hexadecimal Number System and their interconversions. **Binary Arithmetic:** Binary signed numbers, 1's and 2's Complement Arithmetic. **Binary Codes**: Classification, 8421 code, Excess 3 Code, Gray Code, Parity generation and detection. **Logic Gates**: AND, OR, NOT, Universal Gates, XOR and XNOR Gates.	**11**
UNIT-2	**Boolean Algebra:** Logic Operations, Laws of Boolean Algebra, Reducing Boolean Expressions, Introduction to SOP and POS Forms, Boolean Expression and Logic Diagrams, Converting AOI to NAND/NOR Logic. **Minimization of Switching Functions:** 2, 3 and 4 Variable K-map. Don't Care Combinations. **Combinational Logic Design:** Adders, Subtractors, Code converters (Binary to Gray and Gray to Binary), Parity Bit Generator, Comparators, Encoders, Decoders, Multiplexers, Demultiplexers.	**11**
UNIT -3	**Flip-flops**: Latch v/s Flip-Flops- D, JK, RS and T Flip-flop. Master Slave Flip-flops, Flip-flop Excitation Tables. **Shift Registers**: Serial In Serial Out (SISO), Serial In Parallel Out (SIPO), Parallel In Serial Out (PISO) and Parallel In Parallel Out (PIPO) Shift Registers.	**11**

UNIT -4	**Asynchronous Counters**: Ripple Up counters, Ripple Down Counters (Using Positive and Negative edge triggering) **Synchronous Counters**: Design of synchronous counters, Synchronous up counter and Synchronous down counter. Ring Counter and Johnson counter. Applications of counters. **Sequential Circuits:** Design procedure for sequential circuits using state diagrams, state table, state equations, state reduction and assignment. Moore and Mealy Machine.	**12**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TEXTBOOKS:** 1. Anand Kumar, Fundamentals of Digital Circuits, 2nd Edition, PHI Learning, 2007.ISBN-13: 978-8120330607 2. Thomas L. Floyd, Digital Fundamentals, 11th Edition, Pearson Education, 2017.ISBN-13: 978-9332584600 **REFERENCE BOOKS:** 1. M. Morris Mano, Digital Logic and Computer Design, Pearson Education, 2013.ISBN-13: 978-9332542525 2. Albert Paul Malvino & Donald P. Leach, Digital Principles and Applications, 8th Edition, McGraw Hill Education, 2014.ISBN-13: 978-9339203405 3. R.P. Jain, Modern Digital Electronics, 4th Edition, McGraw Hill Education, 2010.ISBN-13: 978-0070681071.	
Course Outcomes:	**Upon completion of the course, students will be able to:** CO 1. Explain the fundamentals of digital and analog systems, number systems, and binary arithmetic CO 2. Analyze and simplify Boolean expressions, and design combinational logic circuits using Boolean algebra and Karnaugh maps. CO 3. Design sequential circuits, including flip-flops, shift registers, and counters. CO 4. Evaluate state-based sequential circuits using Moore and Mealy machines for practical applications.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ECM-203**
Title of the Course : **Logic Design Lab**
Number of Credits : **01**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Nil	
Course Objectives:	This course will enable students to: 1. To know the concepts of Combinational circuits. 2. To understand the concepts of flipflops, registers and counters	
Contents:	**List of Programs /Experiments *(Following experiments should be conducted. A certified journal reporting the experiments conducted should be submitted at the end of the term)***	**No of Hours**
	1. Verification and interpretation of truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates. 2. Realization of logic functions with the help of universal gates NAND and NOR Gate. 3. Realization of Boolean expressions in SOP & POS forms 4. Design of Adders and Subtractors 5. Design of Code Converters 6. Design and Implementation and Verification of Encoders and Decoders 7. Design and Implementation of Multiplexers and Demultiplexers 8. Design and Implement Parity Bit Generators & Comparators 9. Verify the truth table of JK and D flip-flops. 10. Design SISO/SIPO Shift register 11. Design of Synchronous Counter 12. Design of Asynchronous Counter	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**TEXTBOOKS:** 1. Anand Kumar, Fundamentals of Digital Circuits, Fourth Edition, PHI Learning Pvt. Ltd., 2016, ISBN: 9788120352681 2. Thomas L. Floyd, Digital Fundamentals, Eleventh Edition, Pearson Education Limited, 2015, ISBN: 9781292075983 **REFERENCE BOOKS:** 1. Morris Mano, Digital Logic and Computer Design, First Edition, Pearson, 1979, ISBN: 9780132145107 2. Malvino and Leach, Digital Principles and Applications, Fourth Edition, McGraw-Hill, 1986, ISBN: 9780070398832 3. R. P. Jain, Modern Digital Electronics, Third Edition, Tata McGraw-Hill, 2003, ISBN: 9780070494923	

Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Apply logic gate operations to verify and interpret truth tables in digital circuits CO 2. Design combinational circuits like adders, subtractors, code converters, and multiplexers using Boolean algebra and logic gates CO 3. Analyze sequential circuits such as flip-flops, counters, and shift registers to ensure correct functionality CO 4. Design a complete digital system, integrating knowledge of logic gates, combinational and sequential circuits.

Minor/IC/PE

Name of the Programme : B.E in Information Technology
Course Code : ITH 221
Title of the Course : Computer Organisation and Architecture
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic Knowledge of Computer	
Course Objectives:	This course will enable students to: 1. Define fundamental concepts of computer organization and architecture. 2. Understand various number representations and arithmetic operations. 3. Apply knowledge of memory systems and I/O modules to analyze system performance. 4. Analyze processor structures and pipelining to evaluate computer architecture efficiency.	
Contents:		**No of Hours**
Unit - 1	**Introduction:** Organization and Architecture, Structure and function, Computer Components, Computer Function, Interconnection structure, Bus interconnection. **Computer Architecture:** Integer Representation - Unsigned Numbers, Signed Numbers, Signed Magnitude, 2's Complement. **Integer Arithmetic:** Negation, Addition, Subtraction, Multiplication - Unsigned &Signed (Booth's Algorithm), Division. **Computer Arithmetic:** Arithmetic and Logic Unit, Integer Representation, Integer Arithmetic, Floating Point Representation, Floating Point Arithmetic. **Instruction Sets:** Elements of Machine Instruction, Characteristics, Types of Operands, Addressing Modes, Instruction Formats.	**11**
Unit - 2	**Semiconductor Memory:** Memory Hierarchy, Need for Memory Hierarchy, Characteristics of Memory System, Physical Characteristics, Semiconductor RAM **Memories**: Static RAM, SRAM structure, Dynamic RAM, DRAM structure, Difference between SRAM and DRAM, Asynchronous DRAM, Synchronous DRAM, Connection of Memory to the processor, RAM Bus memory, ROM, PROM, EPROM, EEPROM, Flash Memory.	**13**

	Cache Memory: Cache and Main Memory Overview, Cache Organization, Cache Memory Principles, Elements of Cache Design, Unified versus Split Cache. **External Memory**: Magnetic Disk, Floppy Disk, Hard Disk - Read/Write Mechanism, Physical Characteristics, Disk performance parameters, Magnetic Tape, Optical Memory, CD, CD-R, CD-RW, DVD-R, DVD-ROM, RAID, Different RAID Level. **Virtual Memory:** Virtual memory definition, Virtual memory example, Paging, Page Fault, Page replacement Algorithms.	
Unit - 3	**Input/output:** I/O Module, External Devices, I/O External devices block diagram, I/O Functions, I/O Steps, I/O Module Diagram, I/O Mapping**.** **Types of I/O Module:** Programmed I/O, Interrupt Driven I/O, Simple Interrupt Processing, Direct Memory Access (DMA Controller), DMA Configurations, I/O Channel and Processor. **Asynchronous Data Transfer:** Strobe Control and Handshaking, Asynchronous Serial Transfer. **Processor Structure and Functions:** Processor Organization, Register Organization, The instruction cycle. **Instruction Pipeline:** Basic Concepts of Pipelining, two stage instruction pipelining, four stage instruction pipelining Pipeline Performance, Pipeline Hazards: Structural Hazards, Data Hazards, Control Hazards.	**10**
Unit - 4	**Buses:** Bus interconnection structure, traditional bus and high-performance bus architecture, Elements of bus design, Asynchronous v/s Synchronous Buses, PCI Bus, SCSI, USB. **Architecture:** CISC Architecture, Characteristics RISC Architecture, Characteristics, Difference between CISC and RISC. **Multiprocessors:** Characteristics of multiprocessors, Types of Parallel Processor. Interconnection structures: time share common bus, multiport memory, crossbar switch, multistage switching network, hypercube system, **Inter processor arbitration:** serial and parallel. Dynamic arbitration algorithms: Time Slice, Polling, LRU, FIFO, Rotating daisy chain. Symmetric Multiprocessor, Cache Coherence, Multiprogramming v/s Multiprocessing **Control unit operation**: Micro operations, Types of Micro operations, Control of the processor, characteristics of control unit, model of control unit, Micro programmed control: Basic concepts, Microinstruction sequencing, and Microinstruction execution.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	

References/ Readings:	**Text Books** 1. Carl Hamacher, Zvonko Vranesic, Safal Zaky, Computer Organization, 5th Edition, Tata McGraw Hil, 2002, ISBN-13: 978-0070495051. 2. Morris Mano, Computer system architecture, 3rd Edition, Pearson Education, 2017, ISBN-13: 978-9332555610. **Reference Books** 1. Behrooz Parahami, Computer Architecture, 8th Impression, Oxford University Press, 2011, ISBN-13: 978-0199758461. 2. William Stalling, Computer Organization and Architecture: Designing for performance, 11th Edition, Pearson Education, 2022, ISBN-13: 978-0137522679. 3. Kai Hwang, Advanced Computer Architecture - Parallelism, Scalability, Programmability, 2nd Edition, Tata McGraw Hill, 2010, ISBN-13: 978-0070603454.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Identify key components and functions of computer architecture and memory hierarchy. CO 2. Explain number representation schemes and arithmetic algorithms used in computers. CO 3. Demonstrate memory and I/O module operations including cache and virtual memory. CO 4. Compare different processor architectures, pipelining techniques, and multiprocessor systems.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH 222**
Title of the Course : **Computer Organization and Architecture Lab**
Number of Credits : **1**
Effective From AY : **2024-25**

Pre-requisites for the Course:	Basic Knowledge of Computer	
Course Objectives:	The course will enable students to: 1. Understand number systems and floating-point arithmetic 2. Analyze Booth's Algorithm and assembly programs 3. Apply arithmetic and array operations in assembly 4. Develop assembly language programs for given problems	
Content:	**List of Programs**	**No of Hours**
	1. Program to convert a decimal number into binary value and a decimal number into hexadecimal value. 2. Program to Implement Floating-Point Addition. 3. Program to Implement Floating-Point Multiplication. 4. Program to Implement Booth's Algorithm for Two's Compliment Multiplication. 5. Assembly language programming to demonstrate basic arithmetic operations. 6. ALP to reverse a number in an array 7. Program to convert a decimal number into octal value. 8. Program to Implement Multiplication of Unsigned Binary Integers. 9. ALP to calculate factorial & generate Fibonacci series. 10. ALP to calculate sum of Array Elements	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**Text Books** 1. Carl Hamacher, Zvonko Vranesic, Safal Zaky, Computer Organization, 5th Edition, Tata McGraw Hil, 2002, ISBN-13: 978-0070495051. 2. Morris Mano, Computer system architecture, 3rd Edition, Pearson Education, 2017, ISBN-13: 978-9332555610. **Reference Books** 1. Behrooz Parahami, Computer Architecture, 8th Impression, Oxford University Press, 2011, ISBN-13: 978-0199758461. 2. William Stalling, Computer Organization and Architecture: Designing for performance, 11th Edition, Pearson Education, 2022, ISBN-13: 978-0137522679.	

	3. Kai Hwang, Advanced Computer Architecture - Parallelism, Scalability, Programmability, 2nd Edition, Tata McGraw Hill, 2010, ISBN-13: 978-0070603454.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Convert decimal to binary, octal, hexadecimal. CO 2. Illustrate floating-point addition and multiplication algorithms. CO 3. Assess assembly programs for correctness and performance. CO 4. Construct assembly programs for arithmetic and array operations.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-223**
Title of the Course : **Embedded System Design and Applications**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basics of Digital design	
Course Objectives:	This course will enable students to: 1. Recognize the architecture and characteristics of embedded systems. 2. Interpret the role of microcontrollers and memory in embedded applications 3. Differentiate real-time tasks and scheduling algorithms in RTOS. 4. Construct embedded solutions using Arduino and Raspberry Pi.	
Contents:		**No of Hours**
Unit - 1	**Introduction to Embedded System:** Application Domain of Embedded Systems, Desirable features and General Characteristics of Embedded Systems, Model of an Embedded System, Microprocessor vs Microcontroller, Example of a Simple Embedded System, Figures of Merit for an Embedded System, 4/8/16/32 bit MCUs, History of Embedded Systems, current Trends, the Hardware point of View, Microcontroller Unit (MCU), A popular 8-bit MCU, Memory for Embedded Systems, Low power Design, Pullup and Pulldown Resistors.	**12**
Unit - 2	**Real-time Operating Systems:** Real time tasks, Real-time Systems, Types of Real-time Tasks, Real-time Operating Systems, Real-time Scheduling Algorithms, The Earliest Deadline First Algorithm, Qualities of a Good RTOS. **Programming in Embedded C, PIC Programming using MPLAB.**	**11**
Unit - 3	**Prototyping Embedded Devices:** Electronics, Sensors, Actuators, Scaling up the Electronics. **Embedded Computing Basics:** Microcontrollers, System-on-chips, choosing platform. **Arduino:** Developing on Arduino, hardware, Openness, Simple Programs/Projects.	**11**
Unit - 4	**Introduction to Raspberry Pi:** Structure of the boards, Peripherals, Configuring Your PI Linux and Raspberry: Command Line, Linux Commands. **Basic Input and Output:** Using Inputs and Outputs, Digital Output, Digital Input. **Programming inputs and outputs.**	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	

References/ Readings:	**Text Books** 1. Lyla B. Das, Embedded Systems – An Integrated Approach, 1st Edition, Pearson Education, 2011, ISBN-13: 978-8131734211. 2. Matt Richardson & Shawn Wallace, Getting started with Raspberry Pi, 4th Edition, Sharoff/Make Community, 2021, ISBN-13: 978-1680456705. **Reference Books**: 1. McEwen & Hakim Cassimally, Designing the Internet of Things, 1st Edition, John Wiley & Sons Inc, 2013, ISBN-13: 978-1118430626. 2. Simon Monk, The TAB Book of Arduino Projects, 1st Edition, Tab books, 2014, ISBN-13: 978-0071825149. 3. Ruth Suehle, Tom Callaway, Raspberry Pi Hacks, 1st Edition, O'Reilly Media, 2013, ISBN-13: 978-1449318680.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. List key components and application areas of embedded systems. CO 2. Explain features of microcontrollers and memory types. CO 3. Analyze RTOS-based scheduling and real-time system behavior. CO 4. Develop simple embedded applications using standard platforms.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-224**
Title of the Course : **Embedded System Design and Applications Lab**
Number of Credits : **1**
Effective From AY : **2024-25**

Pre-requisites for the Course:	Basics of Digital design	
Course Objectives:	The course will enable students to: 1. Identify components and architecture of Arduino and Raspberry Pi boards. 2. Understand the use of sensors and actuators in embedded system 3. Apply Arduino programming concepts for basic input/output control. 4. Design embedded solutions for real-world applications using microcontrollers.	
Content:	**List of Programs/Experiments**	**No of Hours**
	1. Develop Traffic Light System using Arduino Uno. 2. Develop any application using stepper motor using Arduino Uno. 3. Develop IOT based humidity and temperature monitoring using Arduino Uno. 4. Study of Arduino board 5. Develop blinking LED program using Arduino Uno. 6. Develop switch-based LED control using Arduino Uno. 7. Develop LDR based street light control system using Arduino Uno. 8. Develop digital speedometer using Arduino Uno. 9. Study of Raspberry Pi 5. 10. Mini Project	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**Text Books** 1. Lyla B. Das, Embedded Systems – An Integrated Approach, 1st Edition, Pearson Education, 2011, ISBN-13: 978-8131734211. 2. Matt Richardson & Shawn Wallace, Getting started with Raspberry Pi, 4th Edition, Sharoff/Make Community, 2021, ISBN-13: 978-1680456705. **Reference Books**: 1. McEwen & Hakim Cassimally, Designing the Internet of Things, 1st Edition, John Wiley & Sons Inc, 2013, ISBN-13: 978-1118430626. 2. Simon Monk, The TAB Book of Arduino Projects, 1st Edition, Tab books, 2014, ISBN-13: 978-0071825149.	

	3. Ruth Suehle, Tom Callaway, Raspberry Pi Hacks, 1st Edition, O'Reilly Media, 2013, ISBN-13: 978-1449318680.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. List functionalities of Arduino and Raspberry Pi hardware. CO 2. Explain the integration of sensors and actuators with microcontrollers. CO 3. Implement control logic using Arduino for input/output-based applications. CO 4. Construct an embedded mini-project using development boards.

Multidisciplinary Course

Name of the Programme : B.E. in Information Technology
Course Code : SHM-234
Title of the Course : Engineering Mathematics-II
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic Knowledge of Mathematics	
Course Objectives:	The course will enable the students to: 1. Analyze relations, functions, and integer concepts (prime factorization, GCD, modular arithmetic) 2. Apply propositional calculus, proof techniques, and counting methods. 3. Evaluate vector spaces, linear independence, and rank. 4. Apply analytic geometry concepts and techniques like Gram-Schmidt and matrix decompositions.	
Contents:		**No of Hours**
Unit - 1	**Relations and Functions:** Relations and their properties, Equivalence Relations, partial orderings. One-to-One and Onto Functions, Inverse Function, Composition of functions. **Integers:** Integers and division, primes and greatest common divisors, Euclidean algorithm, Basic properties of Congruence, Modular arithmetic. **Mathematical Induction:** Principle of Mathematical Induction and applications.	**10**
Unit - 2	**Propositional Calculus:** Propositional logic, truth tables, propositional connectives (logical operators), propositional equivalences, propositional implications, complete set of connectives, consistency of statements, theory of inference in propositional logic, predicates and quantifiers. **Counting Principles**: Pigeonhole principle and Inclusion and Exclusion Principle. **Advanced Counting Techniques**: Recurrence relations, formulation, solving linear recurrence relations using characteristic roots.	**12**
Unit - 3	**Linear Algebra**: Vector Spaces, Null and Column Space of a Matrix, Linear Independence, Basis and Rank, Rank-Nullity theorem, Linear Mappings, One-to-One, Onto and Bijective Linear Maps (Isomorphisms), Matrix of a Linear Transformation, Change of Basis Formula. **Analytic Geometry**: Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Gram-Schmidt	**11**

	Orthogonalization, Orthonormal Basis, Orthogonal Complement, Inner Product of Functions	
Unit - 4	**Analytic Geometry:** Orthogonal Projections, Orthogonal Projection and the Normal Equation, Least Squares Problem, Rotations. **Decompositions:** QR Decomposition, Eigen decomposition and Diagonalization via orthogonal Transformation, Cholesky Decomposition, Singular Value Decomposition, Matrix Approximation.	**12**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	1. J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, 1st Edition, McGraw Hill, 2017. 2. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong: Mathematics for Machine Learning, Cambridge University Press, 2020. 3. Swapan Kumar Sarkar, A Text Book of Discrete Mathematics, 9th Edition, S. Chand Publication, 2019. **Reference Books** 1. David C. Lay: Linear Algebra and Its Applications, 5th Edition, Pearson Education India, 2023. 2. Kenneth H. Rosen, Discrete Mathematics and Its Applications, 6th Edition, Tata McGraw Hill, 2006.	
Course Outcomes:	Upon completion of the course the student will be able to: CO 1. Analyze sets, functions, relations, and modular arithmetic for integer computations. CO 2. Solve combinatorial problems using logic and proof techniques. CO 3. Analyze vector spaces and apply linear transformations using matrices. CO 4. Apply analytic geometry and matrix decomposition for multidimensional data analysis	

Skill Enhancement Courses

Name of the Programme : B.E in Information Technology
Course Code : ITH-241
Title of the Course : App Development
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites of the course:	Basic knowledge of programming	
Course Objectives:	This course will enable the students to: 1. Understand a foundational concepts of App Development. 2. Gain proficiency in App Development frameworks and tools. 3. Design and prototype responsive user interfaces. 4. Build and deploy simple applications for real-world scenarios.	
Contents:		**No of Hours**
Unit - 1	**Introduction to App Development:** Overview of mobile and web app development, App development lifecycle, Basics of front-end and back-end development, setting up development environments (Android Studio, React Native CLI, Flutter). **Introduction to app frameworks:** React Native, Flutter, Kotlin, and Swift. **Programming basics for app development:** JavaScript/TypeScript, Dart. Version control with Git and GitHub, Basics of responsive design and mobile-first principles.	**23**
Unit - 2	**Building User Interfaces:** - Understanding UI/UX design principles for mobile apps, designing layouts with Flexbox, Grid, and constraints, Working with widgets, buttons, sliders, and text fields **Navigation design:** Menus, Tab Bars, and Navigation Bars **Incorporating animations:** Transitions, fades, and button animations Accessibility considerations in app design, Working with themes, icons, and colours.	**22**
Unit - 3	**Backend Integration and APIs:** Introduction to databases (SQLite, Firebase), Setting up Firebase for real-time storage, CRUD operations with SQLite and Firestore, Understanding REST APIs and JSON, Fetching, displaying, and sending data using APIs, Authentication and user management using Firebase Auth.	**22**
Unit - 4	**Testing, Debugging, and Deployment:** Debugging tools and error handling techniques. **Writing and running test cases:** Unit testing and integration testing, App optimization: Improving performance and	**23**

	minimizing latency, Security best practices for app development, Preparing apps for demonstrations and feedback. **Mini Project:** Develop and locally deploy a feature-rich app using learned concepts.	
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
Instructions:	**For Unit 1 and Unit 2:** 1. Develop at least two small applications demonstrating the core principles of user interface design and navigation (e.g., a calculator app, to-do list app) showcasing UI layouts and navigation, designed using tools like React Native, Flutter, or Android Studio. 2. Maintain a GitHub repository for code submissions and version control. **For Unit 3 and Unit 4:** 1. Integrate a backend (e.g., Firebase, SQLite, or REST APIs) into at least one app. 2. CRUD operations, authentication, and data retrieval using APIs. 3. Develop a mini-project featuring: Responsive UI design, Database integration **Suggested Software/Tools** • Front-end Tools: Android Studio, React Native CLI, Flutter. • Back-end Tools: Firebase, SQLite, Node.js • Testing Tools: Jest, Flutter Test, Android Studio Debugger	
References:	**Text Books** 1. David Griffiths, Head First Android Development, 2nd Edition, O'Reilly Media, 2017, ISBN -13: 978-1491974056. 2. Marco L. Napoli, Beginning Flutter: A Hands-On Guide to App Development, 1st Edition, Wiley, 2019, ISBN -13: 978-1119550822. **Reference Books** 1. Adam D. Scott, JavaScript Everywhere: Building Cross-Platform Applications with GraphQL, React, React Native, and Electron, 1st Edition, O'Reilly Media, 2020, ISBN -13: 978-1492051569. 2. Fabio Staiano, Designing and Prototyping Interfaces with Figma, 1st Edition, Packt Publishing Limited, 2022, ISBN -13: 978-1801072613. 3. Nader Dabit, React Native in Action, 1st Edition, Manning Publications, 2019, ISBN -13: 978-1617294051.	
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Illustrate key concepts of mobile and web app development, including lifecycle and responsive design. CO 2. Build user-friendly interfaces using tools like React Native, Flutter, and Android Studio. CO 3. Integrate backend services, databases, and APIs for features like CRUD and authentication.	

	CO 4. Test and deploy functional applications across devices, ensuring performance and compatibility.

SEMESTER-IV

Major Courses

Name of the Programme : B.E in Information Technology
Course Code : ITH-204
Title of the Course : Object-Oriented Programming Using Java
Number of Credits : 2
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic knowledge of programming	
Course Objectives:	This course will enable students to: 1. Understand the various concepts of object-oriented programming. 2. Illustrate competency in object-oriented programming by effectively utilizing basic OOP constructs. 3. Apply advanced OOP principles to design and implement applications 4. Develop event driven GUI applications	
Contents:		**No of Hours**
Unit - 1	**Introduction to Java:** Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java. **Objects and Classes:** Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, StringBuffer, File, this reference	**9**
Unit - 2	**Inheritance and Polymorphism**: Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java.	**7**
Unit - 3	**I/O programming:** Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files. **Multithreading in java:** Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try-catch-finally, Collections in java. **Introducing Swing:** Features of Swing, Swing Components: Buttons, Labels, Text Fields, and Containers, Event Handling in Swing, Advanced Swing Components: JScrollPane, JTabbedPane, JTable, and JTree.	**7**
Unit - 4	**Introduction to Java Web Development:** Servlets and JSP Basics (optional, for understanding legacy web apps), Introduction to Spring Boot (Dependency Injection, REST APIs)	**7**

	Database Connectivity in Java: JDBC Basics, Using JPA (Hibernate) for ORM.	
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books** 1. Cay S. Horstmann, Core Java: Fundamentals, 13th Edition, Oracle Pr, 2024. 2. Herbert Schildt and Danny Coward, Java: The Complete Reference, 13th Edition, McGraw Hill Education, 2023. **Reference Books** 1. Joyce Farrell, Java Programming, 10th Edition, Cengage Learning India Private Limited, 2018. 2. Joshua Bloch, Effective Java, 3rd Edition, Addison-Wesley Professional, 2017.	
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Explain the concepts of object-oriented programming. CO 2. Demonstrate competency in object-oriented programming by effectively utilizing basic OOP constructs. CO 3. Apply OOP principles to design and implement applications that solve real- world problems efficiently. CO 4. Develop interactive GUI-based applications.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-205**
Title of the Course : **Object-Oriented Programming Using Java Lab**
Number of Credits : **2**
Effective From AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of programming	
Course Objectives:	The course will enable students to: 1. Understand core OOP concepts and basic Java syntax. 2. Apply inheritance, polymorphism, and multithreading techniques. 3. Develop GUI and web-based applications using Java technologies. 4. Create and manage Java packages, exception handling, and file operations	
Content:	**List of Programs *(Following experiments should be conducted. A certified journal reporting the experiments conducted should be submitted at the end of the term)***	**No of Hours**
	1. Implement basic concepts of OOP: Classes and Objects, Constructors and Overloading 2. Implement different forms of inheritance and demonstrate multiple inheritance using interfaces 3. Implement Java program to demonstrate method overriding and dynamic method dispatch. 4. Implement multithreading using Thread class and Runnable interface. 5. Design a GUI based application using Java Swing controls. 6. Implement a web-based database application. 7. Implement Java programs using decision making and looping statements. 8. Implement java programs using Arrays 9. Implement java programs using Strings 10. Implement Java program to perform file IO 11. Implement Java program that demonstrates the use of Random-Access File class for file IO 12. Implement Java program to demonstrate Generics 13. Implement Java program to create a user-defined package and access it in another program. 14. Implement Java program to create your own exception and handle it. 15. Create a web application using Java servlet.	**60**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	

References/ Readings:	**Text Books** 1. Cay S. Horstmann, Core Java: Fundamentals, 13th Edition, Oracle Pr, 2024. 2. Herbert Schildt and Danny Coward, Java: The Complete Reference, 13th Edition, McGraw Hill Education, 2023. **Reference Books** 1. Joyce Farrell, Java Programming, 10th Edition, Cengage Learning India Private Limited, 2018. 2. Joshua Bloch, Effective Java, 3rd Edition, Addison-Wesley Professional, 2017.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Demonstrate class design, constructors, and method overriding. CO 2. Implement inheritance, interfaces, and multithreading programs. CO 3. Design GUI applications and build simple web database apps. CO 4. Develop Java programs for file I/O, generics, packages, and exception handling.

Name of the Programme : B.E in Information Technology
Course Code : ITH-206
Title of the Course : Algorithm Design and Analysis
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic knowledge of algorithms	
Course Objectives:	This course will enable students to: 1. Understand recursion and algorithm basics. 2. Apply sorting and greedy methods. 3. Develop backtracking and graph algorithms. 4. Assess complexity and NP classes.	
Contents:		**No of Hours**
Unit - 1	**Introduction:** Algorithm Specification, Pseudocode Conventions, Recursive Algorithm, Performance analysis (Space complexity, Time complexity, Asymptotic Notations), Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem method, Comparison of recursion and Iteration. **Divide and Conquer:** General method, Binary search, Finding Maximum and Minimum, Merge sort technique, Quick sort technique	**11**
Unit - 2	**Greedy Method:** General method, Fractional knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees (Prim's & Kruskal's algorithm), Optimal storage on tapes, Optimal merge patterns, Single source shortest paths **Dynamic Programming:** General Method, Multistage Graphs, All pairs shortest paths, Single source shortest path with general weights, Optimal Binary Search Tree, 0/1 knapsack problem, Travelling salesperson problem.	**13**
Unit - 3	**Branch and Bound Method:** General Method, 0/1 knapsack, Travelling salesperson problem. **Backtracking:** General Method, N-queens problem, Sum of subsets problem, graph colouring, Hamiltonian Cycles	**10**
Unit - 4	**Internet Algorithms:** Naïve string-matching algorithm, Rabin Karp algorithm, Knuth-Morris-Pratt algorithm. **Basic Search and Traversal Techniques in Graphs:** Breadth first search, Depth first search, connected components and spanning trees, Biconnected components **NP-hard and NP-complete problems:** Basic concepts, non-deterministic algorithms, NP-Hard and NP- Complete classes	**11**

Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.
References/ Readings:	**Text Books** 1. E. Horowitz, Fundamentals of Computer Algorithms, 2nd Edition, Computer Sci. P, 1984, ISBN-13: 9780812006200. 2. Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, Introduction to Algorithms, 3rd Edition, PHI Learning Pvt. Ltd, 2010, ISBN-13: 9788120340077. **Reference Books** 1. Jon Kleinberg and Éva Tardos, Algorithm Design, 1st Edition, Pearson, 2013, ISBN-13: 9789332553574. 2. Martin C. Brown, Gilles Brassard, Paul Bratley, Fundamentals of Algorithmics, 1st Edition, PHI, 2013, ISBN-13: 9788120344760. 3. Michael T Goodrich and Roberto Tamassia, Wiley VamsiKurama, Algorithm Design: Foundations, Analysis, and Internet Examples, 2nd Edition, Wiley, 2001, ISBN-13: 9780471486480.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Describe recursive algorithms and notations. CO 2. Implement sorting and shortest path algorithms. CO 3. Construct backtracking solutions. CO 4. Differentiate NP-hard and NP-complete problems.

Name of the Programme : B.E in Information Technology
Course Code : ITH-207
Title of the Course : Algorithms Design and Analysis Lab
Number of Credits : 1
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic knowledge of programming	
Course Objectives:	This course will enable students to: 1. Understand recursive sorting techniques. 2. Apply greedy and dynamic programming methods. 3. Develop backtracking and graph algorithms. 4. Evaluate optimization problems using branch and bound.	
Content:	**List of Programs**	**No. of Hours**
	1. Implement merge sort and quick sort using recursion. 2. Construct a minimum spanning tree using Prim's and Kruskal's algorithms 3. Implement the Optimal Binary Search Tree using dynamic programming 4. Implement the 0/1 knapsack problem using the branch and bound strategy 5. Implement the 8-queens problem using backtracking 6. Implement the fractional knapsack problem using the greedy method 7. Implement Dijkstra's algorithm for a given graph 8. Implement finding the shortest distance from a given source to a sink node in a multi-stage graph 9. Implement the sum of subsets problem using backtracking 10. Implement the Hamiltonian cycle problem using backtracking.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References:	**Text Books** 1. E. Horowitz, Fundamentals of Computer Algorithms, 2nd Edition, Computer Sci. P, 1984, ISBN-13: 9780812006200. 2. G.A.V. Pai, Data Structures and Algorithms, 1st Edition, TMH, 2017, ISBN-13: 9781259029536. **Reference Books** 1. Jon Kleinberg and Éva Tardos, Algorithm Design, 1st Edition, Pearson, 2013, ISBN-13: 9789332553574. 2. Martin C. Brown, Gilles Brassard, Paul Bratley, Fundamentals of Algorithmics, 1st Edition, PHI, 2013, ISBN-13: 9788120344760.	

	3. Michael T. Goodrich, Roberto Tamassia, Vamsi Kurama, Algorithm Design: Foundations, Analysis, and Internet Examples, 2nd Edition, Wiley, 2001, ISBN-13: 9780471486480.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Identify appropriate data structures to address specific problem requirements. CO 2. Implement solutions using existing algorithms for defined problems. CO 3. Modify a program to solve a new or unseen problem CO 4. Evaluate the performance of algorithms in terms of time and space complexity.

Name of the Programme : B.E in Information Technology
Course Code : ITH-208
Title of the Course : Fundamentals of Operating System
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic knowledge of computers	
Course Objectives:	This course will enable students to: 1. Understand the concepts of operating system and its services. 2. Analyze process synchronization and explore memory management techniques. 3. Apply the concepts of file system and disk scheduling. 4. Create and execute simple shell scripts.	
Contents:		**No of Hours**
Unit - 1	**Introduction:** Overview of operating systems, OS Operations, Resource Management. **Operating-System Structures:** OS Services, User and OS Interface, System Calls, System services. **Process Management:** Process Concept, Process Scheduling, Operations on Processes, Inter process Communication. **Threads:** Overview, Multithreading Models. **CPU Scheduling:** Basic Concepts, Scheduling criteria, Scheduling algorithms.	**12**
Unit - 2	**Process Synchronization:** The Critical-Section Problem, Mutex Locks, Semaphores. **Deadlocks:** System Model, Deadlock Characterization, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock. **Memory Management:** Background, Contiguous Memory Allocation, Paging, Structure of the Page Table, Swapping Virtual Memory: Background, Demand Paging, Page Replacement.	**12**
Unit - 3	**Storage Management:** Overview of Mass-Storage Structure, Disk Scheduling, Storage Device Management **File System:** File Concept, Access Methods, Directory Structure, Protection, Directory Implementation, Allocation Methods	**11**
Unit - 4	**Shell Programming:** Unix Concepts: understanding UNIX commands, general purpose utilities, file system, handling ordinary files, basic file attributes, VI editor, Basic shell scripts.	**10**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	

References/ Readings:	**Text Books** 1. Silberschatz, Galvin, Gagne, Operating System Concepts, J10th Edition, John Wiley and Sons, 2018, ISBN-13: 9781119456339. 2. Sumitabha Das, Unix Concepts and Applications, 4th Edition, Tata McGraw Hill, 2006, ISBN-13: 9780070587618 **Reference Books** 1. D.M. Dhamdhere, Operating Systems, 3rd Edition, Tata McGraw Hill, 2014, ISBN-13: 9781259003974. 2. Stallings, Operating Systems, Internals and Design Principles, 7th Edition, Pearson Publication, 2012, ISBN-13: 9780132143011 3. Tanenbaum, Modern Operating Systems, 5th Edition, Pearson Publication, 2023, ISBN-13: 9780134743356.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Explain operating system concepts, mechanisms, operations, and shell programming. CO 2. Analyze process management, synchronization, Deadlocks, disk management, File systems in Unix. CO 3. Apply scheduling, synchronization, memory management and shell programming to real time applications. CO 4. Evaluate operating system techniques and create shell scripts for routine tasks.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-209**
Title of the Course : **Fundamentals of Operating System Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of programming	
Course Objectives:	The course will enable students to: 1. Demonstrate CPU scheduling and synchronization problems in operating systems. 2. Simulate deadlock handling and page replacement algorithms. 3. Implement disk scheduling and concurrency control using semaphores. 4. Develop shell scripts and automate tasks with basic shell commands.	
Content:	**List of Programs**	**No of Hours**
	1. Implementation of non-pre-emptive CPU Scheduling algorithms 2. Implementation of pre-emptive CPU Scheduling algorithms 3. Producer-Consumer Problem using Semaphores 4. Simulate algorithm for deadlock prevention and detection 5. Simulate page replacement algorithms: FIFO and LRU 6. Implementation of Simple Shell Scripts 7. Implementation of Dining–philosophers' problem 8. Simulate the algorithm for deadlock avoidance 9. Implementation of Disk Scheduling using FCFS, SCAN algorithm 10. Implementation of Basic Shell Commands	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**Text Books** 1. Silberschatz, Galvin, Gagne, Operating System Concepts, J10th Edition, John Wiley and Sons, 2018, ISBN-13: 9781119456339. 2. Sumitabha Das, Unix Concepts and Applications, 4th Edition, Tata McGraw Hill, 2006, ISBN-13: 9780070587618 **Reference Books** 1. D.M. Dhamdhere, Operating Systems, 3rd Edition, Tata McGraw Hill, 2014, ISBN-13: 9781259003974. 2. Stallings, Operating Systems, Internals and Design Principles, 7th Edition, Pearson Publication, 2012, ISBN-13: 9780132143011 3. Tanenbaum, Modern Operating Systems, 5th Edition, Pearson Publication, 2023, ISBN-13: 9780134743356.	

Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Demonstrate non-pre-emptive and pre-emptive CPU scheduling algorithms. CO 2. Simulate deadlock handling algorithms and page replacement policies. CO 3. Implement synchronization problems and disk scheduling algorithms. CO 4. Create and execute shell scripts and basic shell commands for system tasks.

Name of the Programme : **B.E in Information Technology**
Course code : **ECM-208**
Title of the course : **Engineering Mathematics-III**
Number of Credits : **4**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of Mathematics	
Course Objectives:	The course will enable the students to: 1. Analyze data using descriptive statistics and visual tools. 2. Apply discrete and continuous distributions to compute statistical measures. 3. Conduct point/interval estimation and hypothesis tests across sample sizes. 4. Use linear regression to model relationships and predict outcomes.	
Contents:		**No of Hours**
Unit - 1	**Descriptive Statistics**: Type of data, data collection techniques, Quantitative methods for analysis: Mean, Median, Mode (Empirical formula), Range, Quartiles, Interquartile range, Variance and Standard - deviation, Coefficient of variation, Frequency Distribution and Histogram, Box plot, Scatter diagram.	**10**
Unit - 2	**Elements of Probability Theory**: Sample space & event, Axioms of Probability, Conditional Probability, Bayes' theorem. **Random Variables**: Definition of a random variable, Cumulative distribution function Continuous and Discrete random variables, probability mass function and probability density function, functions of random variables, moment generating function. **Special Distributions**: Bernoulli, Binomial, Poisson, Geometric, Uniform, Exponential and Normal distributions, computation of mean, variance, standard deviation and quartiles, application to numerical problems.	**14**
Unit - 3	**Sampling Theory:** Population & Sampling, standard error, Sampling distribution of Mean (σ known and unknown), Sampling distribution of variance, Point Estimation and Interval Estimation, Alternative and Null Hypotheses, Tests of Hypotheses - one sided and two-sided Hypotheses, Type I and Type II errors. **Test of Hypotheses for large samples**: Test of Hypotheses on single mean and difference of Mean, Test of hypothesis on single proportion and difference of proportions.	**10**
Unit - 4	**Test of hypothesis for small sample**: T-test for single mean and difference of means (equal variances), Chi-square test for Variance and Independence of Attributes, F-test for difference of Variances.	**11**

	Regression Analysis: Method of least squares, Linear regression, correlation coefficient of determination.	
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
Instruction	One or more assignments to be carried out on topics covered in each unit above- Total time allotted for tutorials is 15 Hours.	
References/ Readings:	**Textbooks:** 1. Douglas C. Montgomery George C. Runger, Applied Statistics and Probability for Engineers, 6th Edition, Wiley India Pvt. Ltd., 2017. 2. Richard A. Johnson, Irwin Miller, John Freund, Miller and Freund's Probability and Statistics for Engineers, 9th Edition, Pearson Education India, 2017. **Reference Books** 1. Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, 6th Edition, Academic Press, 2021.	
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Apply descriptive statistics and visualizations (mean, median, mode, variance; histograms, box plots). CO 2. Apply appropriate probability distributions (Binomial, Poisson, Normal) for computing measures. CO 3. Perform point/interval estimation and hypothesis testing for any sample size. CO 4. Analyze relationships and predict outcomes using linear regression.	

Professional Electives

Name of the Programme : B.E in Information Technology
Course Code : ITH-225
Title of the Course : Principles of Computer Graphics
Number of Credits : 3
Effective From AY : 2024-25

Pre-requisites for the Course:	Knowledge of Basic Mathematics	
Course Objectives:	The course aims to enable the students to: 1. Understand the fundamental principles of computer graphics and drawing algorithms. 2. Explore geometric transformations and clipping techniques in 2D graphics. 3. Apply various filling algorithms and construct composite 2D shapes. 4. Design and build simple 3D models using graphics software tools.	
Contents:		**No. of Hours**
Unit - 1	**Overview of graphics systems:** Raster scans systems, Random scan systems. **Output Primitives:** Points and lines, Line drawing algorithms, DDA, Bresenham's line algorithm, Circle generating algorithms, Properties of circles, Midpoint circle algorithm, Ellipse generating algorithm, Properties of Ellipses, Midpoint ellipse algorithm. **Filled area primitives:** Scan line polygon Fill algorithm, inside – outside tests, Scan line fill of curved boundary, Boundary fill algorithm, Flood fill algorithm, Fill area functions.	**12**
Unit - 2	**Two Dimensional Geometric Transformations:** Basic Transformations, Translation, Rotation, Scaling, Composite transformation: Translations, Rotations, Scaling, Other transformations- Reflection, Shear. **Two-Dimensional Viewing:** The viewing pipeline, Viewing coordinate reference frame, Window to viewport coordinate transformation, 2-D viewing functions. **Clipping operations:** Point Clipping, Line clipping, Cohen-Sutherland Line Clipping, Polygon Clipping, Sutherland Hodgeman Polygon clipping, Weiler-Atherton Polygon Clipping, Curve clipping, Text clipping.	**11**
Unit - 3	**Three Dimensional Concepts**: 3-Dimensional display methods, Parallel projections, Perspective projection, Depth cueing, Surface rendering, Exploded and cutaway views.	**11**

	Three Dimensional Geometric and Modeling transformations: Translation Rotation, Coordinate Axes, rotations, Scaling, Reflections, Shears. **Visible surface detection algorithms:** Back – Face detection, Depth buffer method, A – Buffer method, Scan – Line method, Depth Sorting method, BSP- Tree method, Area Sub-division method.	
Unit - 4	**Colour Models and Colour Applications:** Properties of light, Standard primaries, Chromaticity Diagram, XYZ Colour model, CIE Chromaticity Diagram, RGB colour model, YIQ Colour Model, CMY Colour Model, HSV Colour Model, HLS Colour Model. **Computer Animation:** Design of animation sequences, General computer animation functions, Raster Animations, Motion specification, Direct motion specification, Goal directed systems Kinematics and dynamics.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings:	**Textbooks:** 1. Donald Hearn, M. P. Baker, Computer Graphics, 2nd Edition, Prentice Hall of India Pvt. Ltd., 1999, ISBN-13: 978-8120312227. 2. Er. Rajiv Chopra, Computer Graphics (A Practical Approach), Revised Edition, S. Chand publications, 2010, ISBN-13: 978-8121907165. **Reference Books:** 1. Foley, Van Dam, Feiner, Hughes, Computer Graphics: Principles and Practice, 2nd Edition, Addison- Wesley Publishing Company, 1997, ISBN-13: 978-0201848403. 2. N. Krishnamurthy, Introduction to Computer Graphics, 1st Edition, Tata McGraw Hil, 2001, ISBN-13: 978-0074633158. 3. Steven Harrington, Computer Graphics, 2nd Edition, Tata McGraw Hill, 2017, ISBN-13: 978-0070682614.	
Course Outcomes:	Upon completion of the course, the student will be able to: CO 1. Explain raster and random scan systems along with basic drawing algorithms. CO 2. Analyze geometric transformations, viewing, and clipping operations in 2D and 3D CO 3. Implement surface detection techniques and color models in rendering tasks. CO 4. Develop simple animations using motion and color modeling techniques.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-226**
Title of the Course : **Principles of Computer Graphics Lab**
Number of Credits : **1**
Effective From AY : **2024-25**

Pre-requisites for the Course:	Basics of programming	
Course Objectives:	The subject aims to enable students to: 1. Understand basic graphics algorithms including line and circle drawing techniques. 2. Apply 2D geometric transformations and clipping algorithms to graphical objects. 3. Demonstrate various region-filling techniques and shape construction using graphics commands. 4. Develop simple 2D and 3D visual models using appropriate graphics tools and environments.	
Contents:	**List of Programs**	**No of Hours**
	1. Implement Bresenham's line drawing algorithm. 2. Implement Midpoint circle generation algorithm. 3. Implement basic 2D transformations - translation, scaling for a given 2-D object 4. Implement Cohen-Sutherland Line Clipping algorithm. 5. Implement Boundary fill algorithm 6. Implement Flood fill algorithm 7. Implement Digital Differential Analyzer (DDA) line generation algorithm. 8. Program for creating simple 2-D shape of house, car, fish, man using lines, circles using graphics command 9. Implement basic 2D transformations - rotation, shearing and reflection for a given 2-D object 10. 3D Modelling using Blender.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**Textbooks:** 1. Donald Hearn, M. P. Baker, Computer Graphics, 2nd Edition, Prentice Hall of India Pvt. Ltd., 1999, ISBN-13: 978-8120312227. 2. Er. Rajiv Chopra, Computer Graphics (A Practical Approach), Revised Edition, S. Chand publications, 2010, ISBN-13: 978-8121907165. **Reference Books:** 1. Foley, Van Dam, Feiner, Hughes, Computer Graphics: Principles and Practice, 2nd Edition, Addison- Wesley Publishing Company, 1997, ISBN-13: 978-0201848403.	

	2. N. Krishnamurthy, Introduction to Computer Graphics, 1st Edition, Tata McGraw Hil, 2001, ISBN-13: 978-0074633158. 3. Steven Harrington, Computer Graphics, 2nd Edition, Tata McGraw Hill, 2017, ISBN-13: 978-0070682614.
Course Outcomes:	Upon completion of the course, the student will be able to: CO 1. Explain fundamental algorithms used in 2D graphics. CO 2. Analyze geometric transformations on 2D shapes CO 3. Implement line drawing, clipping, and filling algorithms. CO 4. Create simple 2D/3D models using graphics functions and tools.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-227**
Title of the Course : **Computer Networks**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of Data communication	
Course Objectives:	This course will enable students to: 1. Explain basic network models, devices, and protocols. 2. Describe transmission media and data link layer functions. 3. Analyze routing algorithms and congestion control. 4. Summarize transport and application layer protocols.	
Contents:		**No of Hours**
Unit - 1	**Network models and Physical layer:** Define Computer Networks, Types of Networking Devices: Repeaters, Bridges, Switch, Routers, Gateway, Network Interface Card, Hubs, Types of Hubs Layered Task, The OSI Reference Model, TCP/IP protocol Suite, **Basics concepts of communication:** Line configuration, **Types Network Topologies:** Mesh, Star, Tree, Bus and Ring and Hybrid Technologies. **Transmission Modes:** Simplex, half Duplex and Full-Duplex. **Transmission Media:** Guided Media: Twisted pair cable: shielded and Unshielded, Coaxial cable and Optical fibre. **Unguided Media:** Wireless Communication, Terrestrial microwave, satellite communication and cellular telephony. **Transmission Impairments:** Distortion, attenuation and noise, Shannon's Theorem, Comparison of different Media.	**12**
Unit - 2	**Data link layer:** Data link layer Functions, Flow Control – Stop and Wait Flow Control, Sliding Window. **Error Detection:** Types of errors, Detection Methods: VRC, LRC, CRC, Checksum. **Error Control:** Stop and Wait ARQ, Go-Back-N ARQ and Selective-Reject ARQ. **Switching:** Packet Switching, Message Switching and circuit switching, Medium Access Control Sub layer (MAC), the channel allocation Problem, **Multiple Access Protocols:** ALOHA, Types of ALOHAS, Carrier Sense Multiple Access (CSMA) protocols, Collision-free protocol, Bit Stuffing, Bit-Map Binary Countdown, Limited contention protocols, Adaptive Tree Walk Protocol.	**11**

Unit - 3	**Network Layer:** Network Layer functions, Network Layer design issues, **Routing Algorithms:** Optimality principle, shortest path, Flooding, Distance Vector Routing, Link state Routing, **Need for congestion control:** Open loop and Closed loop, Choke Packets, Load Shedding, Jitter Control. **Internet Protocol**: IP Address, Classes, IP ver. 4, IP ver. 6, Difference between IP Ver. 4 and Ver. 6, DHCP, Components of DHCP. Address Resolution Protocol, Reverse Address Resolution Protocol, Internet Control Message Protocol, Types of ICMP error reporting messages, ICMP query messages Internet Group Message Protocol.	**12**
Unit - 4	**Transport Layer:** Transport Layer Functions, UDP, Purpose of UDP, UDP Header, TCP, the TCP Service Model, TCP Segment Header, TCP Connection Establishment, The TCP Connection Release, Comparison of TCP and UDP, Ports and Sockets. **Application Layer:** Domain Name System – DNS, Need for DNS, Types of Domains, FTP, TFTP, Comparison of FTP and TFTP, Telnet Protocol, Hyper Text Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP).	**10**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings:	**Text Books** 1. Andrew S Tanenbaum; Computer Networks, 5th Edition, Pearson Education, 2022, ISBN-13: 978-9332584538. 2. Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, McGraw Hill Education, 2013, ISBN-13: 978-0073376226. **Reference Books** 1. J.S Katre; Computer Network Technology, Tech-Max Publications, 2010. 2. William Stallings, Data and Computer Communication, 10th Edition, Pearson Education, 2017, ISBN-13: 978-0133506488.	
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Explain network models, devices, and layered protocols. CO 2. Analyze data link layer functions and error control methods. CO 3. Evaluate routing algorithms, congestion control, and IP addressing. CO 4. Summarize transport and application layer protocols and their functions.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-228**
Title of the Course : **Computer Networks Lab**
Number of Credits : **1**
Effective From AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of Data communication	
Course Objectives:	The course will enable students to: 1. Demonstrate CRC error detection technique. 2. Configure network topologies and verify connectivity using Packet Tracer. 3. Apply Distance Vector and Link State Routing protocols in network configuration. 4. Develop a basic client-server chat application using socket programming.	
Content:	**List of Programs/Experiments**	**No of Hours**
	1. Implement CRC error detection method. 2. Configure a Network Topology using Packet Tracer Software and ping from any one machine to another machine in the network. 3. Configure a network using Distance Vector Routing protocol with the help of Packet Tracer Software. 4. Create a simple client and server chat application using socket programming. 5. Implement sliding window protocol. 6. Study of different types of Network cables and practically implement any one of them. 7. Study of the basic network devices. (Repeater, Hub, Switch, Bridge, router and Gateway) 8. Implement network IP. (Classification of IP, sub netting and Super netting). 9. Configure of DNS, SMTP, FTP and Web Server. 10. Configure a network using Link State Routing protocol with the help of Packet Tracer Software.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References/ Readings:	**Text Books** 1. Andrew S Tanenbaum; Computer Networks, 5th Edition, Pearson Education, 2022, ISBN-13: 978-9332584538.	

	2. Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, McGraw Hill Education, 2013, ISBN-13: 978-0073376226. **Reference Books** 1. J.S Katre; Computer Network Technology, Tech-Max Publications, 2010. 2. William Stallings, Data and Computer Communication, 10th Edition, Pearson Education, 2017, ISBN-13: 978-0133506488. 3. Jim Kurose, Keith Ross; Computer Networking: A Top-down Approach, 5th Edition, Addison Wesley, 2009, ISBN-13: 978-0136079675.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Analyze network protocols and topologies using simulation tools. CO 2. Apply routing and error detection methods to configure networks. CO 3. Evaluate network device functions and communication integrity. CO 4. Design client-server applications using socket programming.