2025
International Year
of Cooperatives
Cooperatives Build a Better World

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

(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/2025-26/279 Date: 04.08.2025

CIRCULAR

In continuation to the Circular No. GU/Acad –PG/BoS -NEP/2024/508 dated: 12.09.2024, the syllabus for Semester III & IV of the **Bachelor of Engineering in Computer Engineering** 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 Computer Engineering** 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: 2025.08.04 17:09:12 +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 Copmuter 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.

COMPUTER SCIENCE AND ENGINEERING SHEME AY 2024-2025

SEMESTER - III							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**Credits**
1	**Major**	**CMP-200**	Data Structures and Algorithms using C++	3	0	0	3
		CMP-201	Data Structures and Algorithms using C++ Lab	0	0	1	1
		CMP-202	Digital System Design and Analysis	3	0	0	3
		CMP-203	Digital System Design and Analysis Lab	0	0	1	1
2	**Minor/IC/ PE**	**CMP-221**	Computer organization and Architecture	3	0	0	3
		CMP-222	Computer organization and Architecture Lab	0	0	1	1
		OR					
		CMP-223	Microprocessors and Interfacing	3	0	0	3
		CMP-224	Microprocessors and Interfacing 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**	**CMP-241**	Web Technology: Design and Development	0	0	3	3
			TOTAL	**12**	**0**	**8**	**20**

** AEC Courses shall be notified by the University based on the recommendations of respective Board of Studies in languages.*

SEMESTER - IV

Sr. No.	Course Category	Course Code	Title of the Course	L	T	P	Credits
1	Major	**CMP-204**	Object Oriented Programming Systems	2	0	0	2
		CMP-205	Object Oriented Programming Systems Lab	0	0	2	2
		CMP-206	Internet of Things	3	0	0	3
		CMP-207	Internet of Things Lab	0	0	1	1
		CMP-208	Automata Theory and Formal Languages	3	0	0	3
		CMP-209	Automata Theory and Formal Languages Lab	0	0	1	1
		CMP-210	Software Engineering and Project Management	3	0	0	3
		CMP-211	Software Engineering and Project Management Lab	0	0	1	1
2	Professional Elective	**CMP-225**	Graph Theory and Combinatorics	3	0	0	4
		CMP-226	Graph Theory and Combinatorics Lab	0	0	1	
			OR				
		CMP-227	Computational Number Theory	3	0	0	4
		CMP-228	Computational Number Theory Lab	0	0	1	
			TOTAL	**14**	**0**	**6**	**20**

SEMESTER III

Major Courses

Name of the Programme	**: B.E. in Computer Engineering**
Course Code	**: CMP-200**
Title of the Course	**: Data Structures and Algorithms using C++**
Number of Credits	**: 3(3L)**
Effective from AY	**: 2024-2025**

Pre-requisites for the Course:	Programming in C	
Course Objectives:	This course will enable students to: 1. Understand the basic concepts of C++ programming and linked list. 2. Acquire knowledge of stacks, queues, binary trees and their implementation. 3. Construct different types of trees, heaps, graphs and perform different operations on them. 4. Understand the fundamentals of complexity analysis, different types of sorting and hashing.	
Contents:		**No of Hours**
Unit - 1	**A C++ Primer:** Abstract Data Types, Basics of C++ Programming Elements, Expressions, Control Flow, Functions, Encapsulation, Classes, the standard Template Library, Vectors in the STL, C++ Program and File Organization, Writing a C++ Program. **Object- Oriented Design:** Goals, Principles, Inheritance and Polymorphism, Templates, Exceptions. **Linked List:** Singly Linked List: Insertion, Deletion, Search; Doubly Linked List and its Implementation, Circular Linked list and its Implementation, Lists in the standard Template Library.	**11**
Unit 2	**Stacks, Queues:** Stacks, Queues, Priority Queues, Stacks in the STL, Queues in the STL, Priority Queues in the STL, Deques in the STL. **Binary Trees:** Trees, Binary Trees, and Binary search Trees, Implementing Binary Trees, searching a Binary Search trees, Tree Traversal: Breadth - First Traversal, Depth-First Traversal; Insertion, Deletion: By Merging and by Copying.	**11**
Unit 3	**Binary Trees:** Balancing a Tree: AVL Trees; Heaps: Heaps as Priority Queues, Organizing Arrays as Heaps. **Multiway Trees:** The Family of B-Trees: B-Trees, B* Trees, B+ trees. **Graphs:** Graph Representation, Graph Traversals	**11**

Unit 4	**Complexity Analysis:** Computational and Asymptotic Complexity, Big-O Notation, Properties of Big-O Notation, Ω and Θ Notations, The Best, Average, and Worst Cases, Amortized Complexity. **Sorting:** Elementary Sorting Algorithms: Insertion Sort, Selection Sort, Bubble Sort; Efficient Sorting Algorithms: Shell Sort, Heap Sort, Quick sort, Merge sort, Radix Sort, Counting Sort, Sorting in the Standard Template Library. **Hashing:** Hash Functions: Division, Folding, Mid-Square Function, Extraction, Radix Transformation, Universal Hash Functions; Collision Resolution: Open Addressing, Chaining	**12**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
Readings	Text Books: 1. Adam Drozdek, Data structures and Algorithm in C++, Fourth Edition, Cengage Learning. 2012, ISBN 1285415019 2. Michael T. Goodrich, R. Tamassia and David Mount, Data structures and Algorithms in C++, second edition, John Wiley and Sons, 2011, ISBN 0470383275 Reference Books: 1. S. Sahni, Data structures, Algorithms and Applications in C++, 2nd edition, Silicon Press, 2004 ISBN 0-929306-32-5 2. Mark Allen Weiss, Data structures and Algorithm Analysis in C++, Fourth Edition, Pearson Education. Ltd. 2013, ISBN ISBN-13: 9780133404180	
Course Outcomes	The student will be able to CO 1. Explain the fundamentals of C++ on various data structures. CO 2. Implement and analyse stacks, queues and binary Trees. CO 3. Illustrate the differences between various types of trees, heaps and graphs. CO 4. Demonstrate and implement different sorting and hashing techniques.	

Name of the Programme	**: B.E. in Computer Engineering**
Course Code	**: CMP-201**
Title of the Course	**: Data Structures and Algorithms using C++ Lab**
Number of Credits	**: 1**
Effective from AY	**: 2024-2025**

Pre-requisites for the Course:	Programming in C	
Course Objectives:	This course will enable students to: 1. To develop a strong understanding of object-oriented programming concepts, including classes, objects, constructors, destructors, and polymorphism. 2. To learn the implementation and manipulation of various data structures such as linked lists, stacks, queues, trees, and graphs. 3. To apply collision resolution techniques in hashing and understand their role in efficient data storage and retrieval. 4. To analyze and implement sorting algorithms, including both elementary and efficient techniques, to solve real-world problems.	
Contents:		**No of Hours**
Experiment List	1. To implement classes, objects, constructors, destructors, compile time polymorphism. 2. To implement run time polymorphism, inheritance, templates. 3. To implement the various operations such as creation, traversal, searching, insertion, deletion, reverse on a Singly Linked List. 4. To implement the various operations such as creation, traversal, searching, insertion, deletion, reverse on a Doubly Linked List using templates. 5. To implement a Stack to perform a) delimiter matching. b) large number addition. 6. To implement a Queue to check for palindromes, using an array with templates. 7. To implement the collision resolution techniques in hashing and separate chaining. 8. To implement the various operations such as creation, traversals, searching, insertion, deletion, height of a Binary Search tree. 9. To implement Elementary Sorting Algorithms: Selection Sort, Insertion Sort, Bubble Sort. 10. To implement Efficient Sorting Algorithms: Quick Sort, Merge Sort, Heap Sort.	**30**
Pedagogy:	Instructional and Collaborative Learning, Inquiry Based Learning,	

	Experiential Learning.
Readings	**Text Books:** 1. Adam Drozdek, Data structures and Algorithm in C++, Fourth Edition, Cengage Learning. 2012, ISBN 1285415019 2. Michael T. Goodrich, R. Tamassia and David Mount, Data structures and Algorithms in C++, , second edition, John Wiley and Sons, 2011, ISBN 0470383275 **Reference Books:** 1. S. Sahni, Data structures, Algorithms and Applications in C++, , 2nd edition, Silicon Press, 2004 ISBN 0-929306-32-5 2. Mark Allen Weiss, Data structures and Algorithm Analysis in C++, Fourth Edition, Pearson Education. Ltd. 2013, ISBN ISBN-13: 9780133404180
Course Outcomes	The student will be able to CO 1. Demonstrate the ability to design and implement object-oriented programs using concepts like inheritance, polymorphism, and templates. CO 2. Apply operations such as creation, traversal, searching, insertion, and deletion on various data structures, including linked lists, stacks, queues, and binary search trees. CO 3. Implement collision resolution techniques in hashing and graph traversal algorithms using adjacency matrices. CO 4. Implement elementary and efficient sorting algorithms

Name of the Programme : B.E. in Computer Engineering
Course Code : CMP-202
Title of the Course : Digital System Design and Analysis
Number of Credits : 3(3L)
Effective from AY : 2024-2025

Pre-requisites for the Course:	Basic mathematical skill	
Course Objectives:	This course will enable students to: 1. Learn number system conversions, arithmetic, codes and logic gates. 2. Comprehend Boolean expressions and design combinational circuits using logic gates. 3. Understand flip-flops and registers for data storage and sequence control. 4. Learn concept of asynchronous, synchronous counters and basics of sequential circuits.	
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 Compliment 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, Applications. **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, Applications.	**11**
Unit - 4	**Asynchronous Counters:** Ripple Up counters, Ripple Down	**12**

	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:** Finite state model, Memory elements.	
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
Reference Readings:	**Text Books** 1. A. Anand Kumar, Fundamentals of Digital circuits, PHI, Fourth Edition, 2016, ISBN 8120352688 2. Thomas L. Floyd, Digital Fundamentals, Prentice Hall, Eleventh Edition, 2007. ISBN 9353942497 **Reference Books** 1. Morris Mano, Digital Logic and Computer Design, PHI Publication, First Edition, 2016. ISBN 9789332542525 2. Malvino & Leach, Digital Principles and Applications, TMH Publication, First Edition, 2006. ISBN 0028018214 3. R. P. Jain, Modern Digital Electronics, TMH Publication. Fourth Edition, 2009.ISBN 0070669112	
Course Outcomes:	After going through this course, the students will be able to: CO 1. Explain the concepts of number systems, binary arithmetic, codes and logic gates. CO 2. Analyze and simplify Boolean expressions, and design combinational logic circuits using Boolean algebra and Karnaugh maps. CO 3. Illustrate sequential circuits, including flip-flops, shift registers, and counters. CO 4. Apply the concept of asynchronous, synchronous counters and basics of sequential circuits.	

Name of the Programme : B.E. in Computer Engineering
Course Code : CMP-203
Title of the Course : Digital System Design and Analysis Lab
Number of Credits : 1P
Effective from AY : 2024-2025

Pre-requisites for the Course:	Basic mathematical skill	
Course Objectives:	This course will enable students to: 1. Apply number system conversions and arithmetic. 2. Apply Boolean expressions to design combinational circuits using logic gates. 3. Experiment with flip-flops and registers for data storage and sequence control. 4. Implement asynchronous, synchronous counters and sequential circuits.	
Content:	**List of Experiments**	**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. Construction of Half Adder and Full Adder 4. Construction of Half Subtractor and Full Subtractor 5. Design Code Converters (Binary to Gray and Gray to Binary). 6. Realization of BCD Adder 7. Design and Implementation of Parity Bit generator/checker 8. Implementation of Multiplexer & Demultiplexer 9. Verify the truth table of JK and D flip-flops. 10. Design SISO, SIPO Shift register	**30**
Pedagogy:	Inquiry based Learning, Constructive and Collaborative Learning	
Reference Readings:	**Text Books** 1. A. Anand Kumar, Fundamentals of Digital circuits, PHI, Fourth Edition, 2016, ISBN 8120352688 2. Thomas L. Floyd, Digital Fundamentals, Prentice Hall, Eleventh Edition, 2007. ISBN 9353942497 **Reference Books** 1. Morris Mano, Digital Logic and Computer Design, PHI Publication, First Edition, 2016. ISBN 9789332542525 2. Malvino & Leach, Digital Principles and Applications, TMH Publication, First Edition, 2006. ISBN 0028018214 3. R. P. Jain, Modern Digital Electronics, TMH Publication. Fourth Edition, 2009.ISBN 0070669112	

Course Outcomes:	After going through this course, the students will be able to: CO 1. Implement logic gates and verify and the truth tables. CO 2. Implement combinational circuits like adders, subtractors, code converters, and multiplexers using Boolean algebra. CO 3. Verify sequential circuits such as flip-flops, counters, and shift registers to ensure correct functionality. CO 4. Design and implement a complete digital system, integrating knowledge of logic gates, combinational and sequential circuits.

Minor/IC/PE

Name of the Programme : B.E. in Computer Engineering
Course Code : CMP-221
Title of the Course : Computer Organization and Architecture
Number of Credits : 3(3L)
Effective from AY : 2024-2025

Pre-requisites for the Course:	Basics Knowledge of a Computer System	
Course Objectives:	This course will enable students to: 1. Understand the basic concepts and structure of computers. 2. Understand concepts of Microinstructions and microprogramming for instruction cycle and its execution. 3. Explain different types of addressing modes/instruction formats and memory organization. 4. Learn the different configuration of buses, I/O devices and pipelining	
Contents:	Basic Computer awareness	**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.	**11**
Unit - 2	**Internal Memory**: Semiconductor Memory - Memory Hierarchy, Characteristics of Memory System, Semiconductor RAM Memories, Internal Organization of Memory Chip, Static RAM, Asynchronous DRAM, Synchronous DRAM, Connection of Memory to the processor, RAM Bus memory. Cache Memory: Basics of Cache, Structure, Read operation, Elements of Cache Design. Associative Memory: **External Memory:** Magnetic Disk, RAID, Optical Memory. Virtual Memory: Logical VS Physical Address space, working Principle, Mapping Functions, Replacement Policy. operators, Pointers.	**11**
Unit - 3	**Input/output**: I/O Module, External Devices, Programmed I/O, Interrupt Driven I/O, Simple Interrupt Processing, Direct Memory Access (DMA Controller), DMA Configurations, I/O Channel and Processor.	**11**

	Asynchronous Data Transfer: Strobe Control and Handshaking, Asynchronous Serial Transfer. **Processor Structure and Functions**: Architecture of 8085 microprocessor, pin details and functional block diagram of 8085 microprocessor, 8085 programming model, Address, data and control buses, demultiplexing of buses, **Instruction Pipeline**: Basic Concepts of Pipelining, Two stage instruction pipelining, Four stage instruction pipelining Pipeline Performance, Pipeline Hazards: Structural Hazards, Data Hazards, Control Hazards.	
Unit-4	**Buses:** Buses: Bus interconnections, Bus Design Elements, Asynchronous v/s Synchronous Buses, PCI Bus, SCSI,USB,RISC and CISC architectures **Control Unit Operation:** Micro Operations, Control of the CPU, Hardwired Implementation Micro programmed Control: Basic Concepts, Microinstruction Sequencing, and Microinstruction Execution **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.	**12**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
Reading	**Textbooks** 1. William Stalling; Computer Organization and Architecture: Designing for performance; Pearson Education; 2010; 8/e ; . ISBN 978-81-317-3245-8. 2. Ramesh S. Gaonkar, “Microprocessor Architecture, Programming, and Applications with the 8085”, 5th Edition, Penram International; ISBN 8187972882 **Recommended Readings:** 1. Carl Hamacher, Zvonko Vranesic, Safal Zaky; Computer Organization;5/e, 2001, ISBN 978-0071122184 2. Morris Mano; Computer system architecture; Pearson Education;1993;3/e; ISBN 81-7808-687 3. Patterson and Hennessy; Computer Architecture: A Quantitative Approach; Morgan Kaufmann Publishers; 1996; 2/e; ISBN 1-55860-329-8. 4. Kai Hwang; Advanced Computer Architecture - Parallelism, Scalability, Programmability; Tata McGraw Hill, 2010; 3/e.2017 ISBN 9339220927	

Course Outcomes:	The student after undergoing this course will be able to: CO 1. Describe the organization and functionality of the Control Unit, Arithmetic and Logic Unit (ALU), Memory Unit, and I/O Unit. CO 2. Identify and evaluate high-performance architecture designs. CO 3. Analyze techniques for pipelining/instruction-level parallelism to optimize processor performance. CO 4. Design Main Memory modules and study the performance enhancement features of cache memory and I/O techniques.

Name of the Programme : **B.E. in Computer Engineering**
Course Code : **CMP-222**
Title of the Course : **Computer Organization and Architecture Lab**
Number of Credits : **1P**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics Knowledge of Computer, Awareness regarding any programming language like 'C'	
Course Objectives:	This subject equips students with: 1. Understanding of the relationship between hardware and software. 2. Understand how different machine instruction set affects the efficiency of computation. 3. Implementation of different arithmetic operations inside the processor. 4. Explore simulator tool to understand the internal working of the computer system	
Content:	**List of Experiments**	**No. of Hours**
	1. Implement a C program to convert a Hexadecimal, octal, and binary number to decimal number vice versa. 2. Implement a C program to perform Binary Addition & Subtraction. 3. Implement a C program to perform Multiplication of two binary numbers. 4. Implement a C program to perform Multiplication of two binary numbers (signed) using Booth's Algorithms 5. Implement a C program to perform division of two binary numbers (Unsigned) using restoring division algorithm. 6. Implement a C program to perform division of two binary numbers (Unsigned) using non-restoring division algorithm. 7. Write an assembly language code in GNUsim8085 to implement data transfer instruction. 8. Write an assembly language code in GNUsim8085 to store numbers in reverse order in memory location. 9. Write an assembly language code in GNUsim8085 to implement arithmetic instruction. 10. Write an assembly language code in GNUsim8085 to add two numbers using LXI instruction.	**30**
Pedagogy	Inquiry based Learning, Constructive and Collaborative Learning	
References/Reading	**Textbooks** 1. William Stalling; Computer Organization and Architecture: Designing for performance; Pearson Education; 2010; 8/e ; . ISBN 978-81-317-3245-8. 2. Ramesh S. Gaonkar, "Microprocessor Architecture, Programming, and	

	Applications with the 8085”, 5th Edition, Penram International ISBN 8187972882 **Recommended Readings:** 1. Carl Hamacher, Zvonko Vranesic, Safal Zaky; Computer Organization;5/e, 2001, ISBN 978-0071122184 2. Morris Mano; Computer system architecture; Pearson Education;1993;3/e; ISBN 81-7808-687 3. Patterson and Hennessy; Computer Architecture: A Quantitative Approach; Morgan Kaufmann Publishers; 1996; 2/e; ISBN 1-55860-329-8. 4. Kai Hwang; Advanced Computer Architecture - Parallelism, Scalability, Programmability; Tata McGraw Hill, 2010; 3/e.2017 ISBN 9339220927
Course Outcomes:	The student after undergoing this course will be able to: CO 1. Simulate the memory organization of the computer System CO 2. Apply appropriate 8085 instructions for writing assembly programs CO 3. Analyze the different algorithms for the implementation of arithmetic operations inside the computer system. CO 4. Evaluate the performance of the computer system

Name of the Programme	**: B.E. in Computer Engineering**
Course Code	**: CMP-223**
Title of the Course	**: Microprocessors and Interfacing**
Number of Credits	**: 3 (3L)**
Effective from AY	**: 2024-2025**

Pre-requisites for the Course:	Basics Knowledge of Computer	
Course Objectives:	This subject equips students with: 1. Understanding of 8086 Microprocessor Architecture, Instruction Set, Operating Modes and Programming. 2. Use 8086 microprocessor for various applications. 3. Analyse various peripherals for microprocessor based systems. 4. Explore the architectures of different modern processors	
Contents:		**No. of Hours**
Unit-1	**Introduction to 8086 Microprocessor:** 8086 Microprocessor - Architecture, Organization, and Instruction set, Addressing modes, Interrupt system. Pin diagram, Minimum mode 8086 system and timings, Maximum mode 8086 system and timings. Brief description of architectures of 80x86, Pentium, i3/i5/i7 processors	**11**
Unit-2	**Assembly Language Programming:** Assembler directives, Assembly language programs (8086) with Assembler directives for addition, subtraction, multiplication, division etc., sorting and searching, bit manipulation, look-up tables, string manipulations, Macros and Delay subroutines, Debugging	**11**
Unit-3	**Data transfer schemes and Peripheral Interfacing:** Synchronous, Asynchronous, Interrupt driven and DMA type schemes, 8255 PPI and its interfacing, Programmable Communication Interface (8251 USART) and its interfacing, Programmable Interval Timer (8254) and its interfacing, Programmable interrupt controller (8259) and its interfacing, Programmable DMA controller (8257) and its interfacing	**11**
Unit-4	**Memory and I/O Interfacing to 8086**: Address decoding techniques, Interfacing Static RAM and ROM chips, ADC and DAC Interfacing. **Case studies:** Traffic light controller, Stepper motor control, Data acquisition, Temperature measurement and control.	**12**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
Reading	**Text Books** 1. K. Ray and K.M. Bhurchandi, "Advanced Microprocessors and	

	Peripherals", TMH, 2017, ISBN 1259006131 2. Douglas V. Hall, "Microprocessors and Interfacing: Programming and Hardware", 2nd Edition, Tata McGraw-Hill. 1991, ISBN 0070257426 **Reference Books** 1. Barry B. Brey, “The Intel Microprocessors-Architecture, Programming and Interfacing”, 8th Edition, PHI, 2008, ISBN 0135026458 2. Ramesh S. Gaonkar, “Microprocessor Architecture, Programming, and Applications with the 8085”, 5th Edition, Penram International ISBN 8187972882 3. Raj Kamal, Microcontrollers Architecture, Programming, Interfacing and System Design , Pearson Education, 2011. ISBN 8131759903 4. Steve Furbur, ARM System on chipArchitecture,2nd Edition,Addison Wesley, 2000, **ISBN** 0201675196 5. Y.Liu and Glenn A.Gibson, “Microcomputer Systems:8086/8088 Family Architecture, Programming and Design”, 2nd Edition, PHI, ISBN 0135818516
Course Outcomes:	On successful completion of this course, the students will be able to CO 1. Explain various components of the processor and also list out various features of microprocessors and peripherals. CO 2. Describe the internal block diagram of microprocessors and peripherals, addressing modes, instruction set and data transfer schemes. CO 3. Develop algorithm and assembly language programs to solve problems CO 4. Design the microprocessor based system to solve real time problems

Name of the Programme : **B.E. in Computer Engineering**
Course Code : **CMP-224**
Title of the Course : **Microprocessors and Interfacing Lab**
Number of Credits : **1P**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics Knowledge of Computer	
Course Objectives:	This subject equips students with: 1. Understanding the working principles of 8086 Microprocessor. 2. Understanding of 8086 Architecture, Instruction Set, Operating Modes and Programming. 3. Application of 8086 microprocessor for various applications. 4. Analyse various peripherals for microprocessor based systems.	
Contents:	**List of Experiments**	**No. of Hours**
	1. Implement the Byte and word data transfer in different addressing modes. 2. Implement the Block move (with and without overlap 3. Implement Addition and Subtraction of multi precision nos. 4. Implement Multiplication and Division of signed and unsigned Hexadecimal nos. 5. Implement ASCII adjustment instructions 6. Implement Arithmetic programs to find square cube, LCM, GCD, factorial 7. Implement a program to check Logical 1‟s and 0‟s in a given data 8. Implement Finding largest and smallest nos. in arrays, Ascending and descending order 9. Implement Near and Far Conditional and Unconditional jumps, Calls and Returns 10. Implement Programs on String manipulation like string transfer, string reversing, searching for a string, etc.	**30**
Pedagogy:	Inquiry based Learning, Constructive and Collaborative Learning	
Reading	**Text Books** 1. K. Ray and K.M. Bhurchandi, "Advanced Microprocessors and Peripherals", TMH, 2017, ISBN 1259006131 2. Douglas V. Hall, "Microprocessors and Interfacing: Programming and Hardware", 2nd Edition, Tata McGraw-Hill. 1991, ISBN 0070257426 **Reference Books** 1. Barry B. Brey, “The Intel Microprocessors-Architecture, Programming and Interfacing”, 8th Edition, PHI, 2008, ISBN 0135026458 2. Ramesh S. Gaonkar, “Microprocessor Architecture, Programming, and	

	Applications with the 8085”, 5th Edition, Penram International ISBN 8187972882 3. Raj Kamal, Microcontrollers Architecture, Programming, Interfacing and System Design, Pearson Education, 2011. ISBN 8131759903 4. Steve Furbur, ARM System on chipArchitecture,2nd Edition,Addison Wesley, 2000, **ISBN** 0201675196 5. Y.Liu and Glenn A.Gibson, “Microcomputer Systems:8086/8088 Family Architecture, Programming and Design”, 2nd Edition, PHI, ISBN 0135818516
Course Outcomes:	On successful completion of this course, the students will be able to CO 1. Demonstrate the ability to design and conduct 8086 processor experiments, analyze and interpret data CO 2. Design a memory module and I/O module for 8086 based system, component or process as per needs and specification of the 8086 processor CO 3. Demonstrate the skills to use modern engineering tools, such as simulator and microprocessor-based board to analyze problems CO 4. Visualize the behavior of the high-end processor family of 80x86

Multi-disciplinary Courses

Name of the Programme	**: B.E. in Computer Engineering**
Course Code	**: SHM-234**
Title of the Course	**: Engineering Mathematics-II**
Number of Credits	**: 3**
Effective from AY	**: 2024-2025**

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	**11**

	Distances, Angles and Orthogonality, Gram-Schmidt 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 Computer Engineering
Course Code : CMP-241
Title of the Course : Web Technology: Design and Development
Number of Credits : 3
Effective from AY : 2024-2025

Pre-requisites for the Course:	Nil	
Course Objectives:	The subject aims to provide the student with: 1. Ability to design and implement static and dynamic website. 2. Illustration of the implementation of JavaScript for dynamic effects. 3. Ability to choose best technologies for solving web client/server problems. 4. Implementation aspects of server-side technologies like PHP.	
Contents:	**List of Programs /Experiments**	**No. of Hours**
	1. Basic HTML Page Structure Create a simple web page with headings, paragraphs, images, lists, tables, and links. 2. Forms in HTML Design an HTML form with input textfields, checkboxes, radio buttons, dropdowns, submit and reset buttons. Also embed audio and video files into a webpage. 3. HTML5 Semantic Elements & New Attributes • Create a webpage using **semantic elements** like <header>, <nav>, <section>, <article>, <aside>, and <footer> for better structure. • Use **new form attributes** like required, placeholder, autofocus, and pattern for validation. • Add an **email input field** with validation using type="email". • Include a **progress bar** using <progress> and a **meter** for displaying a value range. 4. Basic CSS Styling Apply colors, fonts, borders, and margins to an HTML page using inline, internal and external CSS. 5. CSS Box Model & Flexbox Create a layout using the box model and flexbox for proper alignment. 6. Hover Effects & Animations Add hover effects, transitions, and animations to buttons and	**90**

	images. 7. JavaScript Basics & DOM Manipulation (Covering for loop, while loop, do-while loop, if else, switch, break, continue, exit) 8. Form Validation using JavaScript Implement form validation (e.g., required fields, email, password, and phone number validation) before submission. 9. Event Handling in JavaScript Create a webpage where different events trigger actions. (onclick, onload, onreset, onsubmit, onchange, onselect, onfocus, onblur, mouse and keyboard events, event bubbling.) 10. Understanding GET and POST AJAX Methods To learn and demonstrate the use of GET and POST methods in AJAX to send and retrieve JSON and XML data from the server asynchronously. 11. Sending Multiple JSON Objects via AJAX To send multiple JSON objects in a single AJAX request and to handle multiple objects in the server response. 12. Using XMLHttpRequest with Error Handling To learn how to handle errors in AJAX requests, like network issues and server-side errors, and display error messages dynamically. 13. Dynamic User Input with JSON in AJAX Requests To learn how to gather dynamic data from users and send it as JSON to the server using AJAX for real-time interaction. 14. Basic PHP Programs. Traversing Arrays using Loops and Array Iterators, Built-in Array Functions, file handling functions 15. Form Handling using PHP Implementation of PHP HTTP request methods (Form elements Processing a Web Form, and Validating a Form)	
Pedagogy:	Constructive, Collaborative and Creativity Based Learning	
References/ Readings:	**Text Books** 1. N. P. Gopalan and J. Akhilandeswari; Web Technology: A Developer's Perspective; PHI; ISBN: 978-81-203-5006-9 2 2. DT Editorial Services; Web Technologies Black Book; dreamtechpress; ISBN: 9788177229974 3 3. Kogent Learning Solutions; HTML5 Black Book; dreamtech press; ISBN: 978-93-5004- 4 095-9 4. Lindsay Bassett; Introduction to JavaScript Object Notation; O'Reilly Media; ISBN: 978-1- 491-92948-3 5. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, ASP.NET, XML	

	and Ajax, Black Book, Kogent Learning Solutions Inc. **Reference Books** 1. Smith, Ben; Beginning JSON; Apress; ISBN 978-1-4842-0202-9 2. Ajax: The Definitive Guide, Publisher(s): O'Reilly Media, Inc. 3. Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites, Sixth Edition (Grayscale Indian Edition)by Robin Nixon, Publisher Shroff/O'Reilly.
Course Outcomes:	After going through this course, the students will be able to: CO 1. Analyze and apply the role of languages like HTML and CSS to solve real world problems. CO 2. Develop XML and JSON data using AJAX CO 3. Implement and design Dynamic Webpages using JavaScript. CO 4. Design interactive web pages using PHP.

SEMESTER IV

Major Courses

Name of the Programme	**: B.E. in Computer Engineering**
Course Code	**: CMP-204**
Title of the Course	**: Object Oriented Programming Systems**
Number of Credits	**: 2 (2L)**
Effective from AY	**: 2024-2025**

Pre- requisites for the Course:	Basic General Programming Concepts	
Course Objectives:	This course will enable students to: 1. Apply Java programming fundamentals, including control structures, input/output handling, and object-oriented principles, to develop basic applications. 2. Implement advanced object-oriented concepts such as inheritance, polymorphism, abstract classes, interfaces, and exception handling to create modular and reusable Java applications. 3. Utilize Java's Collection Framework, file handling mechanisms, and multithreading techniques to manage data efficiently and execute concurrent processes. 4. Design interactive graphical user interfaces (GUIs) using Swing and integrate database connectivity using JDBC to develop dynamic and data-driven applications.	
Units	**Contents:**	**No of Hours**
Unit-1	Introduction to Java, First step towards Java Programming, Naming Convention and Data types, Operators in Java, Control statement in java, Input and Output - Accepting input from keyboard, Reading input with java.util.Scanner Class, Displaying output using System.out.printf() and String.format(), Arrays, Strings, Introduction to OOPs, Classes and Objects	**7**
Unit-2	Methods in Java - Static Methods, Static Block, Keyword 'this', Instance method, Passing primitive data types to methods, Passing object/arrays to methods Relationship between Objects, Inheritance, Polymorphism with variables, Methods, Static methods, Private methods, Final methods, Final class Abstract classes, Interfaces, Packages, Exception Handling	**8**
Unit-3	Collection Framework - Collection Objects, Retrieving element from collection, Stack Class, LinkedList class, ArrayList class, Vector class,	**7**

	Streams and Files - Creating file using FileOutputStream, Reading data from a file using FileWriter, Reading a file using FileReader Threads - Single tasking, Multi-tasking, Uses of threads, Creating a thread and running it, Terminating the thread, Single tasking using thread, Multitasking using thread	
Unit-4	Graphics Programming using Swing - JFC, javax.swing and MVC, Window Panes, Creating a Frame in Swing, Displaying Text in Frame, JComponent Class Methods, Creating push button with all features, Displaying Image in Swing, Creating component in swing, Setting look and feel of component, JTable class Java Database Connectivity - Database Servers/Clients, JDBC, Working with Mysql Database, stages in JDBC program, Registering the driver, Connecting to a database, Preparing SQL statements, Using jdbc-odbc bridge driver to connect to oracle database, Retrieving data from Mysql Database	**8**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books** 1. Core Java: An Integrated Approach – *Dr. R. Nageswara Rao*, Dreamtech Press, 2nd Edition, 2016, ISBN 9351199258 **Reference Books:** 1. Java: The Complete Reference – *Herbert Schildt*, McGraw Hill, 12th Edition, 2021, ISBN 1260463419 2. Programming with Java – *E. Balaguruswamy*, McGraw Hill, 6th Edition, 2019, ISBN 9353162335,	
Course Outcomes:	After going through this course, the students will be able to: CO 1. Apply Java programming fundamentals, including data types, control statements, and object-oriented principles, to develop basic applications. CO 2. Implement inheritance, polymorphism, abstract classes, and exception handling to create modular and reusable Java applications. CO 3. Utilize Java's Collection Framework and file handling techniques to efficiently manage data and implement multi-threaded applications. CO 4. Design interactive GUI-based applications using Swing and integrate database connectivity using JDBC for data-driven applications.	

Name of the Programme : B.E. in Computer Engineering
Course Code : CMP-205
Title of the Course : Object Oriented Programming Systems Lab
Number of Credits : 2P
Effective from AY : 2024-2025

Pre- requisites for the Course:	Basic General Programming Concepts	
Course Objectives:	This course will enable students to: 1. Understand and apply Java programming concepts, including data types, control statements, input/output operations, and object-oriented principles, to develop structured programs. 2. Develop and implement advanced object-oriented features such as inheritance, polymorphism, abstract classes, interfaces, and exception handling to build efficient and reusable applications. 3. Explore and utilize Java’s Collection Framework, file handling techniques, and multithreading capabilities to optimize data management and parallel processing. 4. Design graphical user interfaces using Swing components and connect Java applications to databases using JDBC for seamless data interaction.	
Units	**List of Programs /Experiments**	**No of Hours**
Content:	**List of Experiments:** 1. Basic Java Program & Control Structures 2. Arrays Manipulation 3. String Manipulation 4. Classes and Objects 5. Java Methods 6. Inheritance 7. Polymorphism 8. Interfaces 9. Abstract Classes 10. Exception Handling 11. Collection Framework 12. File Handling 13. Multithreading 14. GUI Programming using Swing 15. GUI Application using Database connectivity	**60**
Pedagogy:	Constructive, Collaborative and Creativity Based Learning	
References/ Readings:	**Text Books** 1. Core Java: An Integrated Approach – *Dr. R. Nageswara Rao*, Dreamtech Press, 2nd Edition, 2016, ISBN 9351199258 **Reference Books:**	

	1. Java: The Complete Reference – *Herbert Schildt*, McGraw Hill, 12th Edition, 2021, ISBN 1260463419 2. Programming with Java – *E. Balaguruswamy*, McGraw Hill, 6th Edition, 2019, ISBN 9353162335,
Course Outcomes:	After going through this course, the students will be able to: CO 1. Demonstrate proficiency in Java programming by applying fundamental concepts such as control structures, object-oriented principles, and input/output operations. CO 2. Implement advanced object-oriented techniques, including inheritance, polymorphism, abstract classes, interfaces, and exception handling, to develop modular and reusable applications. CO 3. Utilize Java's Collection Framework, file handling mechanisms, and multithreading concepts to efficiently manage data and execute concurrent processes. CO 4. Design and develop interactive GUI applications using Swing and establish database connectivity using JDBC for creating dynamic and data-driven applications.

Name of the Programme : B.E. in Computer Engineering
Course Code : CMP-206
Title of the Course : Internet of Things
Number of Credits : 3(3L)
Effective from AY : 2024-2025

Pre- requisites for the Course:	Basic computer hardware knowledge like digital circuits and interfacing	
Course Objectives:	This course will enable students to: 1. Understand fundamental concepts and definitions of IoT 2. Apply data handling and analytics tools in IoT. 3. Explain the role of big data, cloud computing and data analytics in a typical IoT system. 4. Understand the role of IoT in various domains of Industry.	
Units	**Contents:**	**No of Hours**
Unit-1	Basics of Networking: Network Types, Layered network models, Emergence of IoT: Evolution of IoT, Enabling IoT and the Complex Interdependence of Technologies, IoT Networking Components	**11**
Unit-2	IoT Sensing and Actuation: Sensors, Sensor Characteristics, Sensorial Deviations, Sensing Types, Sensing Considerations, Actuators, Actuator Types, Actuator Characteristics IoT Processing Topologies and Types: Data Format, Importance of Processing in IoT, Processing Topologies, IoT Device Design and Selection Considerations, Processing Offloading.	**11**
Unit-3	IoT Development Boards: Arduino IDE and Board Types, Raspberry Pi Development Kit, interfacing external gadgets, controlling output, reading input from pins, RFID Principles and components, Wireless Sensor Networks: History and Context, Connecting nodes, Networking Nodes, WSN and IoT.	**11**
Unit-4	Applications of IoT: Home Automation, Smart Cities, Energy, Retail Management, Logistics, Agriculture, Health and Lifestyle, Industrial IoT, Legal challenges, IoT design Ethics, IoT in Environmental Protection.	**12**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books** 1. Vijay Madisetti and Arshdeep Bagga, — “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014, ISBN 0996025510 2. Sudip Misra, Anandarup Mukherjee, Arijit Roy, “Introduction to IoT”, Cambridge University Press 2021 ISBN 978-1108968874 **Reference Books:** 1. Hakima Chaouchi, — “The Internet of Things Connecting Objects to the	

	Web” ISBN : 978-1- 84821-140-7, Wiley Publications 2. Olivier Hersent, David Boswarthick, and Omar Elloumi, — “The Internet of Things: Key Applications and Protocols”, Edition 2, Wiley Publications, 2011, ISBN 1119966701, 3. J. Biron and J. Follett, "Foundational Elements of an IoT Solution", O'Reilly Media, 2016. ISBN 1491950986
Course Outcomes:	After going through this course, the students will be able to: CO 1. Explain the fundamental concepts and definitions of IoT CO 2. Describe the roles of sensors in IoT. CO 3. Demonstrate different protocols used for IoT design. CO 4. Apply different data handling and analytics tools in IoT to develop an application.

Name of the Programme : B.E. in Computer Engineering
Course code : CMP-207
Title of the course : Internet of Things Lab
Number of Credits : 1
Effective from AY : 2024-2025

Pre- requisites for the Course:	Basic computer hardware knowledge like digital circuits and interfacing	
Course Objectives:	This course will enable students to: 1. To study fundamental concepts and definitions of IoT 2. To be familiar with data handling and analytics tools in IoT. 3. Appreciate the role of big data, cloud computing and data analytics in a typical IoT system. 4. Understand the role of IoT in various domains of Industry.	
Units		**No of Hours**
Content:	**List of Experiments:** 1. Understanding Arduino UNO Board and Components. 2. Installing and work with Arduino IDE. 3. Blinking LED sketch with Arduino. 4. Simulation of 4-Way Traffic Light with Arduino. 5. Using Pulse Width Modulation. 6. LED Fade Sketch and Button Sketch. 7. Analog Input Sketch (Bar Graph with LEDs and Potentio metre). 8. Digital Read Serial Sketch (Working with DHT/IR/Gas or Any other Sensor) . 9. Working with Adafruit Libraries in Arduino. 10. Spinning a DC Motor and Motor Speed Control Sketch.	**30**
Pedagogy:	Constructive, Collaborative and Creativity Based Learning	
References/ Readings:	**Text Books** 1. Vijay Madisetti and Arshdeep Bagga, — "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT, 2014, ISBN 0996025510 2. Hakima Chaouchi, — "The Internet of Things Connecting Objects to the Web" ISBN : 978-1- 84821-140-7, Wiley Publications **Reference Books:** 1. Arduino® UNO R3 Manual: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf 2. Matt Richardson & Shawn Wallace; Getting started with Raspberry Pi, O'Reilly Media, 2016, ISBN 1491950986 3. Lyla B.Das, Pearson India; Embedded Systems – An Integrated Approach; Pearson Education India, 2012, ISBN 9332511675,	

Course Outcomes:	**After going through this course, the students will be able to:** CO 1. Design a basic IoT model CO 2. Implement applications using sensors. CO 3. Demonstrate different protocols used for IoT design. CO 4. Apply different data handling and analytics tools in IoT to develop an application.

Name of the Programme : **B.E. in Computer Engineering**
Course Code : **CMP-208**
Title of the Course : **Automata Theory and Formal Language**
Number of Credits : **3(3L)**
Effective from AY : **2024-2025**

Pre-requisites for the Course:	Basics Knowledge of Computer	
Course Objectives:	The students will be able to: 1. Learn the mathematical definition of a finite automaton, including states, transitions, input alphabet, start state, and accepting states 2. Understand the differences between DFAs and NFAs, how to design and analyze each type, and the conversion between them 3. Recognize that finite automata can only recognize regular languages and learning how to express regular languages using regular expressions 4. Understand what types of problems cannot be solved using finite automata and the need for more powerful computational models for complex tasks.	
Contents:		**No. of Hours**
Unit-1	Sets, Logic, Functions, Relations, Mathematical Induction. Introduction to Languages, Grammars and Automata. Chomsky Hierarchy. Finite Automata: Deterministic Finite Accepters, Nondeterministic Finite Accepters, Equivalence of Deterministic and Nondeterministic Finite Accepters, Reduction of the Number of States in Finite Automata.	**11**
Unit-2	Regular Languages and Regular Grammars: Regular Expressions, Connection Between Regular Expressions and Regular Languages, Regular Grammars, Closure properties of Regular languages, A Pumping Lemma for regular languages. Finite State Transducers: Mealy Machine, Moore Machine, Moore and Mealy Machine Equivalence	**11**
Unit-3	Context-Free Languages: Examples of Context Free Languages, Leftmost and Rightmost Derivations, Derivation Trees, Parsing and Ambiguity, Methods for Transforming Context Free Grammars, Chomsky Normal Form, and Greibach Normal Form. Nondeterministic Pushdown Automata, Pushdown Automata and Context-Free Languages, Deterministic Pushdown Automata, Pumping Lemma for Context-Free Languages. Closure of Context Free languages.	**11**
Unit-4	Turing Machine: Standard Turing Machine, Combining Turing `s for Complicated Tasks, Turing's Thesis. Turing Machines with More Complex Storage. Nondeterministic Turing Machines. A	**12**

	Universal Turing Machine. Linear Bounded Automata. Computability and Decidability: Turing Machine Halting Problem. Unrestricted Grammars, Context Sensitive Grammars.	
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings:	**Text Books:** 1. Peter Linz; An introduction to Formal Languages and Automata; Jones & Bartlett Learning, 2006, ISBN 0763737984 2. John C Martin; Introduction to languages and the theory of computation; Tata McGraw Hill, Fourth Edition, 2010, ISBN 0073191469 **Recommended Readings**: 1. John E. Hopcraft and Jeffery D. Ullman; Introduction to Automata Theory, Languages and Computation; Narosa Publishing House, 2008, ISBN 9788131720479 2. Michael Sipser; Introduction to Theory of Computation; PWS Publishing Company, 2012, ISBN 113318779X. 3. A.A Puntambekar; Formal Languages and Automata Theory; Technical Publications Pune, 2019, ISBN 978-9333200943 4. K.L.P Mishra, N. Chandrasekaran; Theory of Computer Science – Automata, languages and Computation; PHI Publications; Third Edition ; 2008, ISBN 9788120329683	
Course Outcomes:	The student after undergoing this course will be able to: CO 1. Identify formal language classes , properties of languages, grammars and Design the automata CO 2. Apply the techniques to transform between equivalent deterministic and non-deterministic finite automata and regular expressions. CO 3. Design grammars for different language classes, perform simplification of CFG and explain concept of Pushdown automata. CO 4. Explain concept of Turing machine and Construct the Turing recognizable languages.	

Name of the Programme : B.E. Computer Engineering
Course Code : CMP-209
Title of the Course : Automata Theory and Formal Languages Lab
Number of Credits : 1P
Effective from AY : 2024-2025

Pre- requisites for the Course:	Basics Knowledge of Computer, Awareness regarding any programming language like 'C' or C++	
Course Objectives:	The students will be able to: 1. Understand basics of a finite automaton, including states, transitions, input alphabet, start state, and accepting states 2. Design and analyse DFA and NFA types, and the conversion between them 3. Understand to recognize a finite automata defined using a regular expression. 4. Understand different and powerful computational models for complex tasks.	
List	**List of Experiments**	**No of Hours**
	1. Write a program to implement Finite Automata 2. Write a program to implement Deterministic Finite Automata 3. Write a program to implement Non-Deterministic Finite Automata 4. Write a program to implement Mealy State Machine 5. Write a program to implement NFA, DFA, Mealy Machine using JFLAP 6. Write a program to implement Push Down Automata 7. Write a program to implement Context Free Grammar 8. Design a Turing Machine which will increment the given binary number by 1. 9. Design a Turing Machine that calculate 2's complement of given binary string. 10. Case Study on Applications of Finite Automata	**30**
Pedagogy:	Constructive, Collaborative and Creativity Based Learning	
References/ Readings:	**Text Books:** 1. Peter Linz; An introduction to Formal Languages and Automata; Jones & Bartlett Learning, 2006, ISBN 0763737984 2. John C Martin; Introduction to languages and the theory of computation; Tata McGraw Hill, Fourth Edition, 2010, ISBN 0073191469 **Reference Books:** 1. John E. Hopcraft and Jeffery D. Ullman; Introduction to Automata	

	Theory, Languages and Computation; Narosa Publishing House, 2008, ISBN 9788131720479 2. Michael Sipser; Introduction to Theory of Computation; PWS Publishing Company, 2012, ISBN 113318779X. 3. A.A Puntambekar; Formal Languages and Automata Theory; Technical Publications Pune, 2019, ISBN 978-9333200943 4. K.L.P Mishra, N. Chandrasekaran; Theory of Computer Science – Automata, languages and Computation; PHI Publications; Third Edition 2008, ISBN 9788120329683
Course Outcomes:	The student after undergoing this course will be able to: CO 1. Apply basic concepts of formal languages of finite automata techniques. CO 2. Design Finite Automata’s for different Regular Expressions and Languages CO 3. Derive context free grammar for various languages CO 4. Implement the solutions to applying normal form , PDA and Turing Machines

Name of the Programme : **B.E. Computer Engineering**
Course Code : **CMP-210**
Title of the Course : **Software Engineering and Project Management**
Number of Credits : **3(3L)**
Effective from AY : **2024-2025**

Pre- requisites for the Course:	Basic Programming skills	
Course Objectives:	The students will be able to : 1. Understand basic concepts of Software Engineering and analyze various phases of Software Development Life Cycle (SDLC). 2. Learn Object oriented Software Engineering Concepts. 3. Identify Software Quality parameters and understand Software testing practices. 4. Evaluate importance of Software Project Management with its methods.	
Content:		**No. of Hours**
Unit - 1	**Introduction to Software Engineering: Scope of software engineering** - Historical aspects, Economic aspects, Maintenance aspects, Specification and design aspects, Team programming aspects. **The Software Process** - Client, Developer and User Phases of SDLC Life Cycle, Requirement phase, Specification phase, Design phase, Implementation phase, Integration phase, Maintenance phase, **Software Life Cycle Models** Build and Fix Model, Waterfall, Rapid Prototyping Model, Incremental Model, Extreme Programming, Synchronize and Stabilize Model, Spiral Model, Object Oriented Life Cycle Model. **Software Metrics** Capability Maturity Model. **Estimating Duration and Cost** Metrics for size of product, Techniques for cost estimation and models, **Teams**: Team Organization, Democratic Team Approach, Classical chief Programmer Team Approach, Synchronize and Stabilize Teams	**12**
Unit - 2	**Object Oriented Software Engineering:** Object Oriented System Development, Object Oriented Terminology, Types of Cohesion, Types of Coupling, Data Encapsulation, Software re-usability, Portability, Interoperability, CASE tools in use for Object Oriented Software Engineering. **Requirement Phase:** Techniques for Requirement Elicitation and Analysis Metrics for requirement phase, Testing. **Specification Phase:** Specification Document, Metrics for Specification Phase, Testing. **Analysis Phase:** OO Analysis, Use case Modeling, Class Modeling,	**11**

	Dynamic Modeling, Testing **Design Phase:** Action oriented Design and Abstraction, DFA, Data Oriented Design, Object Oriented Design, Testing.	
Unit - 3	**Software Quality Assurance:** Quality Concepts, Quality Movement, Software Reviews, Formal Technical Reviews, Formal Approaches to SQA, Statistical SQA, Software reliability, SQA Plan. **Software Testing:** Fundamentals, test Case Designs, White Box Testing, Basic Path Testing, Control Structure Testing, Black Box Testing, Testing for specialized environment **Software Testing Strategies:** Strategic Approach to Software Testing, Strategic Issues, Unit Testing, Integration Testing, Validation Testing, Organizational approaches to testing, Software testing tools – for classical engineering and object oriented engineering, Software testing standards **Object Oriented Testing**	**11**
Unit - 4	**Software Project Management:** Managing software project, Project planning process planning – Standard process, Requirement change management, Risk management, Configuration management. Project execution, Project monitoring and control, Project closure. Performing closure analysis, closure analysis report.	**11**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
Reading	**TEXT BOOKS** 1. Stephen R. Schah, Object Oriented and Classical Software Engineering- (TMH), 2010, ISBN 0073376183 2. Pankaj Jalote, Software Project Management in practice, Pearson Education, 2015, ISBN 935286882X **REFERENCES** 1. Roger S. Pressman, Software Engineering – A practitioner's approach , McGraw Hill, 2023, ISBN 9355325045 2. Watts S. Humprey, A discipline for Software Engineering , Pearson Education, 2007, ISBN 0201546108 3. K. K. Aggarwal and Yogesh Singh, Software Engineering , New Age Publications, 2008, ISBN 8122423604	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Translate end-user requirements into software requirements and specifications. CO 2. Generate high level design from software specifications. CO 3. Implement and integrate the software, adhering to best practices CO 4. Develop test cases for effective testing of the software	

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-211**

Title of the Course : **Software Engineering and Project Management Lab**

Number of Credits : **1**

Effective from AY : **2024-2025**

Pre- requisites for the Course:	Basic Programming skills
Course Objectives:	**The students will be able to :** 1. Analyze various phases of Software Development Life Cycle (SDLC). 2. Understand Object oriented Software Engineering Concepts. 3. Apply Software Quality parameters and understand Software testing practices. 4. Evaluate importance of Software Project Management with its methods.
Content:	**List of Experiments** Do the following 10 experiments/ exercises for a mini project. 1. Development of problem statement. 2. Identification & justification of process model for software development. 3. Preparation of software requirement specification (SRS) document. 4. Study and development of design documentation. 5. Validation of design documentation. 6. Implementation of project and Integration of modules. 7. Development of testing strategies and test cases for the project. 8. Conduction of unit tests for various modules of project and documentation. 9. Conduction of various tests (including integration test) for project and documentation. 10. Preparation of Final Project Report and user Manuel.
Pedagogy:	Constructive, Collaborative and Creativity Based Learning
Reference/ Readings	**TEXT BOOKS** 1. Stephen R. Schah, Object Oriented and Classical Software Engineering- (TMH), 2010, ISBN 0073376183 2. Pankaj Jalote, Software Project Management in practice, Pearson Education, 2015, ISBN 935286882X **REFERENCES** 1. Roger S. Pressman, Software Engineering – A practitioner's approach , McGraw Hill, 2023, ISBN 9355325045 2. Watts S. Humprey, A discipline for Software Engineering , Pearson Education, 2007, ISBN 0201546108 3. K. K. Aggarwal and Yogesh Singh, Software Engineering , New Age Publications, 2008, ISBN 8122423604

Course Outcomes:	Upon completion of the course, students will : CO 1. Implement end-user requirements into software requirements and specifications. CO 2. Design a software model from a software specification. CO 3. Evaluate the software model for productivity CO 4. Test the software model using test cases.

Professional Electives

Name of the Programme : B.E. Computer Engineering
Course Code : CMP-225
Title of the Course : Graph Theory and Combinatorics
Number of Credits : 3(3L)
Effective from AY : 2024-2025

Pre-requisites for the Course:	Basic Mathematics	
Course Objectives:	The subject aims to provide the student with 1. Understanding of the structure of graphs. 2. Understanding and knowledge of application of the fundamental concepts in graph theory. 3. Application of graph theory-based tools in solving practical problems. 4. Evaluate the specific proof techniques to prove results in graph theory	
Contents:		**No of Hours**
Unit-1	**Basic graph theory Concepts**: Graphs, isomorphism, subgraphs, matrix representation, degree sequence. Bipartite graphs, line graphs, chordal graphs. **Trees**: Characterization, number of trees, Minimum spanning trees	**11**
Unit-2	**Connected Graphs and Shortest Paths:** Walks, trails, paths, connected graphs, distance, Eulerian and Hamiltonian graphs, cut vertices, cut edges, blocks, weighted graphs, shortest paths algorithms, Dijkstra's and Floyd Warshall algorithms	**11**
Unit-3	**Independent sets, coverings and matchings:** Basic equations, matching in bipartite graphs, perfect matching, greedy and approximation algorithms. **Vertex Colouring:** Chromatic number and cliques, Greedy colouring algorithms.	**11**
Unit-4	**Directed Graphs:** Directed Graphs, underlying Graphs, out degree, in degree, connectivity, orientation, Eulerian directed graphs, Hamiltonian directed graphs, tournaments	**12**
Pedagogy	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
Reading	**Text Books:** 1. Graph theory with applications, J.A. Bondy and U.S.R.Murthy, Edition 2, 1977 ISBN 0444194517 2. Introduction to graph theory, D.B.West, Cambridge University Press, Edition ISBN 0132278286	

	Reference Books 1. Graph theory, R.Diestel, Springer, Elsevier Science Publishing, ISBN 9783662701065
Course Outcome	At the end of the course the student will be able to CO 1. Explain induced subgraphs, cliques, matchings, covers in graphs and determine whether graphs are Hamiltonian and /or Eulerian. CO 2. Demonstrate central theorems about trees, matching, connectivity, coloring, and planar graphs. CO 3. Apply basic graphs algorithms to real life problems. CO 4. Synthesise the graph theory as a modeling tool.

Name of the Programme : B.E. Computer Engineering
Course Code : CMP-226
Title of the Course : Graph Theory and Combinatorics Lab
Number of Credits : 1
Effective from AY : 2024-2025

Pre-requisites for the Course:	Discrete Mathematics and basic programming skill
Course Objectives:	The subject aims to provide the student with 1. Implementation of the structure of graphs. 2. Application of the fundamental concepts in graph theory. 3. Use of graph theory-based tools in solving practical problems. 4. Demonstrate the important results in graph theory
Content:	**List of Experiments**
	Implement any 8 of the following experiments. 1. Verify graph isomorphism. 2. Generate various subgraphs. 3. Determine the graph for a given degree sequence. 4. Convert a given graph into a line graph. 5. Find the minimum spanning tree for the given graph. 6. Implement shortest path algorithm. 7. Generate closed walks, trails and paths in a connected graph. 8. Identify a Eulerian circuit in a graph. 9. Identify a Hamiltonian circuit in a graph 10. Implement the graph colouring algorithm
Pedagogy:	Constructive, Collaborative and Creativity Based Learning
References/ Reading	**Text Books:** 1. Graph theory with applications, J.A. Bondy and U.S.R.Murthy, Edition 2, 1977 ISBN 0444194517 2. Introduction to graph theory, D.B.West, Cambridge University Press, Edition ISBN 0132278286 **Reference Books** 1. Graph theory, R.Diestel, Springer, Elsevier Science Publishing, ISBN 9783662701065
Course Outcomes:	At the end of the course the student will be able to CO 1. Verify the concepts of induced subgraphs, cliques, matchings, covers in graphs. CO 2. Determine whether graphs are Hamiltonian and /or Eulerian. CO 3. Demonstrate the applications of trees, matching, connectivity, colouring, and planar graphs. CO 4. Apply some basic algorithms for graphs.to justify graph theory as a modelling tool.

Name of the Programme : B.E. Computer Engineering
Course Code : CMP-227
Title of the Course : Computational Number Theory
Number of Credits : 3(3L)
Effective from AY : 2024-2025

Pre- requisites for the Course:	Discrete Mathematics, and Programming Skills	
Course Objectives:	This course will enable students: 1. To understand fundamental number theory concepts. 2. To apply modular arithmetic and number theory principles 3. To establish cryptographic principles by number theory concepts. 4. To apply number theory algorithms in Classical Cryptosystems.	
Units	**Contents:**	**No of Hours**
Unit-1	Basic Number Theory: Divisibility, Prime numbers, Division algorithm, well ordering principle, principle of induction, Greatest Common Divisor, Euclidean algorithm, Extended Euclidean Algorithm, Coprime integers, LCM, Linear Diophantine Equations, prime number theorem, Goldbach conjecture, twin prime conjecture	**11**
Unit-2	Congruence, Linear congruences, Division, Chinese Remainder Theorem, Modular Exponentiation, Fermat's Little Theorem, Euler 's function, Euler's Theorem, Primitive Roots, Wilson 's Theorem, Quadratic Residues, Inverting Matrices Mod n, Square Roots Mod n, Legendre and Jacobi Symbols, Finite Fields.	**11**
Unit-3	Pseudo-random Bit Generation, LFSR Sequences, Enigma. Primality Testing: Fermat's Primality Test, Miller-Rabin Primality Test, Solovay- Strassen Primality Test. Factoring: p-1 Factoring Algorithm, Quadratic Sieve, Pollard's Rho Method, Discrete Logarithms: Computing Discrete Logs, The Pohlig-Hellman Algorithm.	**11**
Unit-4	Applications of number theory in Classical Cryptosystems: Shift ciphers, Affine Ciphers, The Vigenère cipher, Substitution Ciphers, Sherlock Holmes, The Playfair and ADFGX Ciphers. Symmetric and asymmetric key cryptography, public key cryptosystems: RSA, Diffie Hellmann key exchange.	**12**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings:	**Text Books** 1. Neal Koblitz, A course in Number Theory and Cryptography, Springer, 2006 ISBN 1461264421 2. D. Burton, Elementary Number Theory, McGraw-Hill, 2012 ISBN	

	9355325126 3. Behrouz A. Forouzan, D. Mukhopadhyay, Cryptography and Network Security, McGraw Hill, 2015. ISBN 9339220951 **Reference Books:** 1. J. Pipher, J. Hoffstein and J.H. Silverman, An introduction to Mathematical Cryptography, Springer-verlag 2014. ISBN 1493917110 2. Ivan Niven, H.S. Zukermann, H.L. Montgomery, An introduction to theory of numbers, Willey, 2015 ISBN 8126518111
Course Outcomes:	After going through this course, the students will be able to: CO 1. Develop the mathematical skills to solve number theory problems and to apply the mathematical skills for divisions, primes, and number functions. CO 2. Solve linear, quadratic and system of linear congruence equations. CO 3. Investigate applications of number theory and the use of computers in number theory with methods like discreete logs, factorization and primality testing etc. CO 4. Apply the concepts of number theory in classical cryptosystems.

Name of the Programme : B.E. Computer Engineering
Course Code : CMP-228
Title of the Course : Computational Number Theory Lab
Number of Credits : 1
Effective from AY : 2024-2025

Pre- requisites for the Course:	Discrete Mathematics, and Programming Skills	
Course Objectives:	This course will enable students to: 1. To learn about the use of divisibility theorem 2. To gain knowledge about congruences 3. To understand and learn about cryptosystem 4. To study the applications in industry	
Content:	**List of Experiments:**	**No. of Hours**
	1. Program based on basic Integer Arithmetic. 2. Program on Euclid's Algorithm, Extended Euclid's Algorithm. 3. Program on Chinese Remainder Theorem. 4. Program on Groups, rings and finite fields. 5. Program on various factorization techniques 6. Program to implement various primality testing. 7. Program on pseudorandom bit generation 8. Program to implement Pollard rho method 9. Program to implement Pohlig-Hellman method. 10. Program to implement various ciphers	**30**
Pedagogy:	Constructive, Collaborative and Creativity Based Learning	
References/ Readings:	**Text Books** 1. Neal Koblitz, A course in Number Theory and Cryptography, Springer, 2006 ISBN 1461264421 2. D. Burton, Elementary Number Theory, McGraw-Hill, 2012 ISBN 9355325126 3. Behrouz A. Forouzan, D. Mukhopadhyay, Cryptography and Network Security, McGraw Hill, 2015. ISBN 9339220951 **Reference Books:** 1. J. Pipher, J. Hoffstein and J.H. Silverman, An introduction to Mathematical Cryptography, Springer-verlag 2014. ISBN 1493917110 2. Ivan Niven, H.S. Zukermann, H.L. Montgomery, An introduction to theory of numbers, Willey, 2015 ISBN 8126518111	
Course Outcomes:	After going through this course, the students will be able to: CO 1. Apply number theory concepts to cryptography. CO 2. Solve some of the divisor problems. CO 3. Understand the importance of Euler‟s phi function in RSA crypto	

	system CO 4. Understand the importance of larger primes in coding theory.