BCSE- 2nd Year, 1st Semester

Course code	FET/BS/B/Math/T/211
Category	Basic Science
Course title	**Mathematics III**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – I
Pre-requisites (if any)	

Syllabus:

Probability and Statistics: Definition of probability; Conditional probability and independence; Bayes‘ theorem; Collection and Representation of Statistical data: Measures of Central Tendency & Dispersion; Correlation and Regression; Expectation and Variance; Random variables; Discrete and Continuous distribution; Poisson, Normal and Binomial distribution; Chebysheffs inequality.

Vector Algebra: Basics of vector algebra; Dot and Cross products of two vectors; Product of three or more vectors; volume of tetrahedron; Work done; Moment; Angular velocity. Applications to mechanics.

Vector Calculus: Vector functions of a scaler variable; Limit; Continuity and Derivative of vector functions; Applications to mechanics; Partial derivatives of vector function of more than one variables; Directional derivative; Gradient; Divergence and Curl; Vector Integration; Line integrals; Surface integrals and volume integrals; Greeen’s theorem in the plane; Gausss Theorem; Stokes’ Theorem and their application; Tangent Normal and Binormal of space curve; Serret-Frenet formulae; Normal plane, Rectifying plane and oscillating plane.

Ordinary Differential Equations: First order differential equations - exact, linear and Bernoulli’s form, second order differential equations with constant coefficients, method of variation of parameters, general linear differential equations with constant coefficients, Euler s equations, system of differential equations.

Partial Differential Equations: First order PDE; Lagrange method; Second order PDE with constant coefficients and their classification to Elliptic, Parabolic and Hyperbolic type. Solution of PDE by method of separation of variables; Solution of one-dimensional wave and diffusion equation; Laplace equation of two dimensions.

Course code	CSE/PC/B/T/212
Category	Professional Core
Course title	**Digital Logic and Circuits**
Scheme and Credits	L–T–P: 4-0-0; Credits: 4.0; Semester – I
Pre-requisites (if any)	

Syllabus:

1. Number systems. Number base conversion, Binary codes for decimal digits and code conversion. Error detecting codes- Parity and Hamming codes. Other coding systems - Seven segment code, Alpha Numeric codes like ASCII, EBCDIC, ISCII and Unicode. **[4L]**

2. Digital Arithmetic: Addition and subtraction of unsigned binary numbers. Complement arithmetic; n’s complement and (n-1)’s complement. Representation of signed binary numbers; sign-1’s complement and sign-2’s complement, Addition and subtraction of signed binary numbers. Other binary arithmetic- BCD, NBCD, Excess-3 BCD. **[6L]**

Boolean Algebra: Truth table, logic operations- AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR. De Morgan's theorem. Minimization of Boolean functions - Karnaugh Veitch map method and Quine-McCluskey's method. Digital Logic Gates. **[6L]**

3. Combinational logic circuit design: Half-adder, Full-adder, Encoder, Decoder, Multiplexer, de-multiplexer, parity generator, parity checker, priority encoder, magnitude comparator. **[8L]**
4. Sequential logic circuit design: Flip-flops - SR, JK, Master slave JK, D and T. Registers- serial-in-serial-out, serial-inparallel-out, parallel-in-serial-out, shift registers, circulating shift registers and their applications. Counters - Synchronous, asynchronous, up, down and modulo-n. Finite state machines (FSM) - state table, state diagram, Mealy and Moore machines, state minimization, implementation with flip-flops. **[12L]**
5. Different logic families- Diode Logic, DCTL, RTL, IIL, DTL, HTL, TTL, ECL, MOS & CMOS – their operations, characteristics and specifications. **[8L]**
6. Timing circuits- 555 timer & its use as monostable and astable. **[2L]**
7. Memory devices: semiconductor main memory RAM, ROM, EPROM, EAPROM etc. Secondary storage device principles. **[6L]**
8. Analog digital interfacing: Different D/A and A/D conversion techniques **[4L]**

Books:

1. Digital Logic and Computer Design by M. Morris Mano
2. Digital Principles and Applications by Leach, Malvino and Saha

Course Outcomes (COs):

At the end of the course a student shall be able to:

CO1: Understand various number and coding systems and digital arithmetic

CO2: Understand the minimization of Boolean functions

CO3: Design Sequential and Combinational logic circuit

CO4: Understand the characteristics and operations of different logic families and timing circuits.

CO5: Understand the working principles of memory devices and analog digital interfacing.

Digital Logic and Circuits	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3												1			
CO2	3	2	2										1			
CO3		2	3											3		
CO4		2	3											3		
CO5	2		3											3		

Course code	CSE/PC/B/T/213
Category	Professional Core
Course title	**Data Structures and Algorithms**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – I
Pre-requisites (if any)	

Syllabus:

1. Information, Data, Data Types, Abstract Data Type (ADT), Data Structure, Static and Dynamic Data Structures, Implementation Methods [CO1]
2. Array as an ADT, Single and Multidimensional Arrays, Structures, ADT Polynomial, Sparse Matrix and List using Arrays. [CO1, CO2]
3. Pointers, ADT Linked List, Singly Linked List, Doubly Linked List, Multi-linked List. Implementations using Pointers and Arrays. Application in implementing Polynomials, Sparse Matrix, etc. [CO1, CO2, CO4]
4. Algorithm Design Methodologies – Divide & Conquer, Greedy Algorithms, Dynamic Programming, Backtracking, Exhaustive Search, Probabilistic Algorithms. [CO3, CO4]
5. Analysis of Algorithms, Big O notation, Introduction to analysis of Sequential, Iteration and Recursive Algorithms with examples. Measurement of program efficiency. [CO3, CO4]
6. Development, Implementation, Analysis and Measurement of Searching and Sorting Algorithms – Linear and Binary Search, Insertion Sort, Selection Sort, Merge Sort, Quick Sort, Heap Sort, Counting Sort. [CO3, CO4, CO5]
7. ADT Stack and Queue – Implementation using Arrays and Pointers, Priority Queue, Applications. [CO1, CO2, CO4, CO5]
8. ADT Tree, Binary Tree, Binary Search Tree, Height Balanced Tree, 2-3 Tree, B-Tree,. Applications. [CO1, CO2, CO4, CO5]
9. ADT Graph, Representations of Graph Data Structures, Graph Algorithms – Depth-First and Breadth-First Search, Spanning Tree – Kruskal and Prim's Algorithm, Finding Minimum Cost Paths, Applications.
10. ADT Hash Table – Hash Functions, Synonyms, Collisions, Example Hash Functions, Collision Resolution Strategies, Applications. [CO1, CO2, CO4, CO5]
11. Advances Topics – B+ Tree, Bloom Filters, Applications. [CO1- CO5]

Books:

1. Fundamentals of Data Structures in C – Horowitz, Sahni, Anderson-Fred, Latest Edition. (* Textbook)
2. Data Structures and Algorithm Analysis in C by Mark Alan Weiss, 2nd ed., Pearson Education (#)
3. Data Structues and Algorithms by Aho, Hopcroft & Ullman
4. Data Structures and Program Design by Kruse et. al., PHI
8. Algorithms + Data Structures = Programs by N. Wirth, PHI (#)
9. How to solve it by Computers by Dromey, PHI (#)

A few chapters will be referred

Course Outcomes (CO):

The students of this course should be able to

CO1 Understand Data Type Abstraction for Linear and Non-linear Data Arrangements K2
CO2 Implement Static and Dynamic, Linear and Non-linear Data Structures K3-K5
CO3 Design algorithms using elementary algorithm design strategies K5
CO4 Analyze and implement algorithms and data structures for time and memory usage efficiency K5, K6
CO5 Solve computing problems by selecting / developing data structures and algorithms for efficient implementation K3-K6

Data Structures and Algorithms	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	3	2	2	1					1		1	3	2		1
CO2	3	2	3	2	1					1		1	3	1		1
CO3	3	3	3	3	1					1		1	3			1
CO4	3	3	3	3	1					1		1	3	1		1
CO5	3	3	3	3	1					1		2	3	2		1

Course code	CSE/PC/B/T/214
Category	Professional Core
Course title	**Programming Fundamentals and Object oriented Concepts**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – I
Pre-requisites (if any)	

Syllabus:

Review of Procedural Programming Using C:

Pointers and Functions: function prototype, nested function calls, ***Recursion***, concepts of pointers, Parameter Passing-call by value and call by address, passing arrays to function, void pointer, static vs. Dynamic memory allocation, array of pointers, string handling using pointers, pointer to a function, function returning pointer. [8L]

Structure: nested structure, array of structures, pointer to structure, self – referential structure [3L]

IO Handling: File pointer, File reading, Writing, text mode files, binary mode files [3L]

Pre-processing directives and macro, bit-wise operations [2L]

Object Oriented Concepts and C++:

Introduction to object oriented programming concept [1L]

Overview of Procedural Feature: Concept of Reference variable, Default Parameters to Function, Function overloading, Inline function, Macro [2L]

Fundamental Object Oriented Features:

Structure and Class and Object, Abstraction/ Encapsulation, Access Specifier [1L]

Static Members, Friend Function, Constructor and Destructor, Operator Overloading , Inheritance [6L]

Abstract Class, Run time polymorphism, Virtual Base Class, case studies for class design [3L]

File Handling, Case Studies for class design [4L]

Exception Handling [1L]

Class Template and Function Template [2L]

Introduction to Namespace [1L]

Introduction to STL, Case studies for class design [4L]

Programming Fundamentals and Object Oriented Concepts (COs:)

CO1: Understand and utilize the concept of structure, pointer and functions in C. [K2, K3]
CO2: Develop programs for file processing in C. [K2, K3]
CO3: Understand and realize OOP features through C++ [K1, K2]

CO4: Design and implement the object oriented solution for problems using C++ [K3, K4]
CO5: Understand and utilize the concept of namespace and STL classes in C++ [K2, K3]

		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Programming Fundamentals and Object Oriented Concepts	CO1	3	2	2	2				1	1			2
	CO2	2	1	3	1				1	2			2
	CO3	3	2	2	1				1	1		1	2
	CO4	2	2	3	1	1			1	2	1	2	2
	CO5	1	2	2	3	2			2	1		2	2

		PSO1	PSO2	PSO3	PSO4
Programming Fundamentals and Object Oriented Concepts	CO1	3		1	2
	CO2	3		1	2
	CO3	3		1	2
	CO4	3		1	2
	CO5	3		1	2

Course code	CSE/PC/B/T/215
Category	Professional Core
Course title	**Computer Organisation**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – I
Pre-requisites (if any)	

Syllabus:

Fundamentals of Computers: Introduction to Digital Computers Hardware and Software & their dual nature Von-Neumann Concept, Role of Operating System and Compiler **[1L]**

Instruction Set: **[4L]**

Opcode and operand, Instruction formats, Addressing modes and effective address calculation Instruction decoding and Instruction execution cycle

Arithmetic Unit: **[8L]**

ANSI representation of data, Signed addition and subtraction, Fast addition, carry-look-ahead adders and carry save adders, Different multiplication techniques for signed numbers, Booth's Technique including Bit-pair technique, Binary division techniques. Restoring type and Non-restoring type

Floating point arithmetic and different rounding techniques

Memory Organization: **[8L]**

Memory Hierarchy and different access techniques, Main memory and Secondary memory concepts Memory Interleaving, S-access and C-access organization, Cache Memory, Different mapping techniques and Replacement Algorithms Virtual memory and implementation using Page map table

Control Unit Design: **[8L]**

Instruction interpretation and execution Hardwired control design, Micro-programmed control design Instruction format design and nano-programming

Input Output Organization: **[7L]**

I/O interface and drivers, Programmed I/O, Synchronous and Asynchronous I/O transfer Interrupt driven I/O transfer, Direct Memory Access (DMA), I/O processor

Pipeline Processing: **[4L]**

Suggested Readings:

1. Computer Organization, Hamacher et.al.
2. Computer Architecture & Organization , J.P. Hayes
3. Computer System Architecture, Morris Mano
4. Computer Organization & Design, P. Pal Chaudhuri

Course outcomes (COs): At the end of this course, each student should be able to:

CO1: Develop the ideas about Digital computer's Hardware and Software, role of Operating System and compiler. [K3]

CO2: Illustrate the Instruction formats, Addressing modes, Instruction decoding and Instruction execution cycle. [K2]

CO3: Describe ANSI representation of data, Addition, Subtraction, Multiplication techniques, Restoring type and Non-restoring type, Floating point arithmetic, rounding techniques [K2]
CO4: Explain Memory Hierarchy and different access techniques, Main memory, Secondary memory and Cache Memory concepts, Different mapping techniques and Replacement Algorithms, Virtual memory [K2]
CO5: Investigate different aspects of Instruction interpretation, and design issues of Hardwired and Micro-programmed control design, and nano-programming [K4]
CO6: Develop the concepts of I/O interface and drivers, Synchronous and Asynchronous I/O transfer, Interrupt driven I/O transfer and Direct Memory Access [K5]

Computer Organization	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO4
CO1	3					1						1	2	2		1
CO2	2	3	1										1	3		
CO3	3	2	2	2									1	3		
CO4	2				3								1	3	1	
CO5	3	1	2		2								1	3		1
CO6	3		2		2								1	3		

Course code	CSE/PC/B/S/211
Category	Professional Core
Course title	**Data Structures Lab**
Scheme and Credits	L–T–P: 0-1-2; Credits: 2.0; Semester – I
Pre-requisites (if any)	

Syllabus:

This Lab course comprises of a series of assignments on Data Structures and Algorithms to be implemented in C language. The assignments will follow the progress in the Theory course. They will include implementation of ADTs, Algorithms and applications to solve real-life problems. Measurement of performance of the programs developed will also be part of the assignments.

Details of Assignment types:
1. Problems to be solved as a review of programming in C:
   - Random number and String Generation
   - File handling
   - Non-recursive and recursive versions of Factorial and Fibonacci numbers
   - Computing with large numbers and strings, Linear and binary search
3. Problems on Arrays and Lists, ADT Polynomial and Sparse Matrix, Use of different features of compilation and debugging
4. Problems on various types of Linked Lists and applications
5. Problems on various algorithm development techniques, measurement of timings of programs
6. Problems on Sorting Algorithms, Plotting of timings with increasing number of inputs, reconciliation with theoretical results
7. Problems and applications of Stack and Queue, Demonstration of Modular programming
8. Problems and applications binary trees and balanced trees, mini project for use of modular programming technique
9. Implementation of Graph algorithms and their applications
10. Design and implementation of Hash algorithms, their applications and experimental performance verification
11. Development of Bloom Filters, Persistent Data Structures

Course Outcomes (CO):

The students of this course should be able to

- CO1. Implement a given ADT as a data structure in C language
- CO2. Implement a given Algorithm developed using a design strategy in C language
- CO3. Choose the appropriate data structure and algorithm design method for a specified application to be developed in structured and modular form
- CO4. Apply systematic testing and debugging methods on developed applications and measure the performance of them
- CO5. Write Reports in acceptable format for the applications developed

Data Structure Lab	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	2	3	2	1				2	1		1	3			1
CO2	3	2	3	2	1				2	1		1	3			1
CO3	3	3	3	3	1				2	1	1	1	3		1	2
CO4	3	3	2	3	1				2	1	1	1	3			3
CO5	2	1	1	2	1				1	1		1	3		1	3

Course code	CSE/PC/B/S/212
Category	Professional Core
Course title	**Digital Systems Lab**
Scheme and Credits	L–T–P: 0-0-2; Credits: 1.5; Semester – I
Pre-requisites (if any)	

Syllabus:

Experiment 1: Implement a decimal to binary encoder and a binary to decimal decoder using logic gates
Experiment 2: Implement a 4-to-1 multiplexer and a 1-to-4 demultiplexer.
Experiment 3: Implement a comparator circuit for two 3-bit numbers.
Experiment 4: Implement of an odd/ even parity generator.
Experiment 5: Implement 3-bit adder/subtractor circuit.
Experiment 6: Implement a modulo-n asynchronous counter.
Experiment 7: Implement a modulo-n synchronous counter.
Experiment 8: Study of 555 timer circuits.
Experiment 9: Design a 4-bit ROM using diodes and resistors.
Experiment 10: Implement a DAC using resistive divider circuit.

Course Outcomes (COs):

At the end of the course a student shall be able to:

CO1: Design different combinational logic circuits like adder, subtractor, encoder, decoder, multiplexer, comparator
CO2: Design sequential logic circuits, like, registers, counters.
CO3: Understand the operational principles and applications of timing circuits.
CO4: Understand different techniques of D/A and A/D conversion.

Digital System Lab	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO1 2	PSO1	PSO2	PSO3	PSO4
CO1	2		3											3		
CO2	2		3											3		
CO3	2		3		1									3		
CO4	2		3		1									3		

Course code	CSE/PC/B/S/213
Category	Professional Core
Course title	**Programming Practice Lab**
Scheme and Credits	L–T–P: 0-0-2; Credits: 1.5; Semester – I
Pre-requisites (if any)	

Syllabus:
1. Problems on the use of array, structure and pointers (in C language)
2. Modular programming in C -- use of functions
3. File handling in C
4. Problems on basic object oriented features like encapsulation, overloading, constructor, destructor, inheritance, polymorphism, virtual base class, abstract class, generic class, generic function (using C++)
5. Problems on STL (using C++)
6. Problems on object oriented design of systems and implementation of the same (using C++)

Course Outcomes (COs):

CO1 Able to understand the usage of pointers in C and its applications
CO2 Able to understand and implement OOP features through C++ Programming
CO3 Understand and utilize STL classes in C++
CO4 Able to effectively choose programming components that efficiently solve computing problems in real-world.

CO-PO Mapping (3 – Strong, 2 – Moderate and 1 – Weak)

		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Programming Practice Lab	CO1	2			1	1							
	CO2	1			1	3							
	CO3			1		2	1						
	CO4	1	3	1		1				1			

CO-PSO Mapping (3 – Strong, 2 – Moderate and 1 – Weak)

		PSO1	PSO2	PSO3	PSO4
Programming Practice Lab	CO1	3			
	CO2	1			1
	CO3	3			
	CO4	1	1		

Second Year- Second Semester

Course code	CSE/BS/B/Math/T/221
Category	Basic Science
Course title	**Mathematics IV**
Scheme and Credits	L–T–P: 4-0-0; Credits: 4.0; Semester – I I
Pre-requisites (if any)	

Syllabus:
Discrete Structure:
Set Theory: Review of set theory basics, Partially ordered sets, Lattice, Relations, Equivalence relations and induced partitions, Countable and uncountable sets and their properties. Reordered sets. Least upper bound property. Statement of real number system as an ordered field with least upper bound property. Rational

numbers. Algebraic and transcendental numbers. Infinite decimal expansion of real numbers. Cantor's diagonalisation method for uncountability of real numbers. 10 L

Introduction to Mathematical Logic: Propositions and compound propositions, Basic logical operations, Truth tables, Tautologies and contradictions, logical equivalence, logical implication, inference, quantifiers 6 L

Functions; mappings; injection and surjections; composition of functions; inverse functions; special functions; recursive function theory; 4 L

Proof strategies and Mathematical Induction 2 L

Pigeonhole principle , Permutation and combinations 4 L

Probability Theory and Statistics:
Mathematical Theory of Probability: Basic concepts, Classical and axiomatic approaches, Sample space and events, Properties of probability functions. 4 L

Conditional probability and independent events, Concept of random variable, Discrete and continuous probability density, mass and distribution functions 4L

Expectations and moments, Moment generating and characteristic functions, Uniform, binomial, poisson, exponential and normal distributions, Multi – dimensional random variables and random vectors, Joint, marginal and conditional probability distributions 10L

Functions of random variable and random vector, Linear transformation of random variable and random vector, Independent random variables, Mean square estimation, Correlation and regression, Central limit theorem. 6 L

Introduction to stochastic processes: Markov, stationary and ergodic processes, Correlation function and power spectral density. Introduction to Queuing Theory: Kendall's Notations, M/M/1, M/M/m Queue, effect of bulk arrival 8L

Books:

1. C. L. Liu, Elements of Discrete Mathematics
2. J.L. Matt, A. Kandal and T. P. Taluk Dar: Dicrete Mathematics for Computer Scientists and Mathematicians
3. S.K. Mapa, Higher Algebra, Abstract and Linear
4. Amritava Gupta, Groundwork of Mathematical Probability and Statistics
5. A. M. Goon, M.K. Gupta and B. Dasgupta, Basic Statistics
6. J. Medhi, Stochastic Process
7. R. A, Fisher, An Introduction to Probability theory and its applications, Vol-1

Course Outcomes (CO):

Students will

1. be able to learn the basic mathematical objects such as sets, relations, and mappings and their simple properties.
2. learn basic concepts of real number system including least upper bound property, different representation of real number, and Canforce method for uncountability of real number.
3. be familiar with mathematical logic in the capacity of propositional logic and predicate logic. Gain knowledge in using mathematical induction in basic combinatorics to apply in counting finite elements.

4. understand key concepts of probability including discrete and random variables, probability distribution. will learn general properties of joint marginal and conditional distribution, expectations, moments and variant which help to analyze these distributions.
5. be able to define and explain different popular distribution (normal, binomial, poisson). Be able to understand functions of random variables and random vectors with their linear transformation, correlation, regression, central limit theorem which are relevant to data analysis.
6. be familiar with markov process, correlation functions and power spectral density and understand basic concepts of queuing theory and some important queuing models.

Mathematics IV	Program Outcomes											Program Specific Outcomes			
CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3	PSO4
CO 1	2	1											2		1	
CO 2	2	1											2		1	
CO 3	2	1		2									2		1	
CO 4	1			2									1		2	
CO 5	1			2									1		2	
CO 6	1			2									1		2	

Course code	CSE/PC/B/T/222
Category	Professional Core
Course title	**Advanced Object Oriented Programming**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – II
Pre-requisites (if any)	

Syllabus:

Object Oriented Programming with JAVA:

Introduction to Java: Properties of Java, JVM. [1L] [CO1]

Primitive data types and strings. [2L] [CO2]

- Representation of numeric data (integers, floating point, characters, strings)
- Wrapper classes

Standard operations on primitives and strings. [2L] [CO2]

- Variables, types, expressions, and assignment

Object-Oriented Programming Concepts. [10L] [CO3]

- Classes, Objects, Methods, Constructors, Accessors, Mutators etc.
- Packages, interfaces
- Inheritance
- Compile time and Run time Polymorphism

Exception handling. [2L] [CO4]
Handling exceptional situations via algorithms isolated for that purpose
Exception classes

Concurrency-- Threads and Synchronization [2L] [CO5]
Input/Output (I/O) [2L] [CO6]
Text file I/O, Binary file I/O- Class object I/O with object streams
Graphical User Interfaces (GUIs). [2L] [CO7]

- Standard GUI components (buttons, text fields, text areas)
- Action Events and Action Listeners

Dynamic Data Structures. [4L] [CO6]

- Collection API

Python *Programming [24L]*

Introduction to Python	[1L]	[CO1]
Fundamentals and Pre-requisites	[1L]	[CO1]
Basic Components	[4L]	[CO2]
Functions	[2L]	[CO2]
Strings Revisit	[2L]	[CO2]
Lists and Dictionaries	[2L]	[CO2]
OOP Concepts:		
Classes and Objects	[2L]	[CO3]
Inheritance & Polymorphism	[2L]	[CO3]
Operator Overloading	[1L]	[CO3]
Exceptions and Error Handling	[1L]	[CO4]
Applications:		
Regular Expressions	[1L]	[CO2]
Files and Data Structures	[1L]	[CO2]
Multi-threaded Programming	[1L]	[CO5]
Network Programming	[1L]	[CO5]
Database Programming	[1L]	[CO5]
Text Processing	[1L]	[CO7]

Reference Books:

1. Introduction to computing and problem solving using Python, E. Balagurusamy, McGraw Hill
2. Programming and problem solving with Python, Ashok Namdev Kamthane & Amit Ashok Kamthane, McGraw Hill
3. Progress in Computer Science with Python, Sumita Arora, Dhanpat Rai and Co.
4. Computer Science with Python, Sumita Arora, Dhanpat Rai and Co.
5. Core Python Programming, R. Nageswara Rao, Dremtech Press
6. Core Python Applications Programming, Wesley J. Chun, Pearson
7. Python Programming – A Modular Approach, Sheetal Taneja & Naveen Kumar, Pearson
8. Python - The compete reference , Martin C. Brown, McGraw Hill
9. Python Programming using Problem Solving Approach, Reema Thareja, Oxford

Course Outcomes (CO):

The students of the course should be able to –

CO1 Be familiar with platform independent programming concepts
CO2 Be exposed to primitive and reference data types in Java and Python

CO3 Able to apply object oriented features of Java to problem solving
CO4 Be exposed to exception handling, Threads and concurrent programming in Java and Python
CO5 Be familiar with designing user interfaces, event driven programming
CO6 Exploit API support for solving problems and apply knowledge in related applications

CO-PO Mapping (3 – Strong, 2 – Moderate and 1 – Weak)

		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Advanced Object Oriented Programming (Java and Python)	CO1	1											
	CO2					2							
	CO3	1		2		1							
	CO4			1	2								
	CO5				1		2						
	CO6	1		3									

CO-PSO Mapping (3 – Strong, 2 – Moderate and 1 – Weak)

		PSO1	PSO2	PSO3	PSO4
Advanced Object Oriented Programming (Java and Python)	CO1			2	
	CO2	1			1
	CO3			2	
	CO4	3			
	CO5	2			
	CO6	3			

Course code	CSE/PC/B/T/223
Category	Professional Core
Course title	**Microprocessors and Assembly Language Programming**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – II
Pre-requisites (if any)	

Syllabus:

Introduction to microprocessor , Basic features of 8085 microprocessors and its addressing modes, 8085 microprocessor architecture [2L]

Memory and I/O interfacing , Address decoding, Address aliasing, Memory read and write operations, Timing diagrams, Memory mapped I/O and I/O mapped I/O [4L]

Programming of 8085 Instruction Set, Assembly Language Programming and Illustrative examples [6L]

8085 Interrupt Structure
Data Transfer Techniques [2L]

Synchronous and Asynchronous modes of data transfer, Interrupt driven I/O, DMA [2L]
Peripheral Devices, 8255 programmable peripheral interface, 8254 programmable counter, 8251 UART programmable communication interface, 8257 DMA Controller. 8259 Interrupt controller, 8279

Keyboard & display interface. Signal converter and their interfacing techniques- ADC 0809, DAC 0808. [10L]

Introduction to micro-controller, 8051 as an example. Micro-controller architecture, bi-directional data ports, internal ROM and RAM, counters/timer s, oscillator and clock, serial communication., 8051-register set, memory organization – internal & external, program memory & data memory, bit addressable memory, and special function registers , Introduction to instruction set of 8051 and assembly language programming , Introduction to advanced microprocessors – (contd.) [8L]

8086 as an example, 8086Architecture and Internal Resister Set, Brief discussion on Instruction Set, Min-Max mode, Concept of Co-processor and its interfacing, Brief studies on Important features of higher processor in the Intel 80X86 family. [6L]

Suggested Readings:

1. R. Gaonkar, “Microprocessor Architecture, Programming and Applications”, 5th Ed., Pearson International, 2001.
2. K. Ayala, “The 8051 Microcontroller – Architecture, Programming and Applications”, 2nd Ed., Pearson International,1996.
3. Liu and Gibson, “Microcomputer Systems: The 8086/8088 Family”, 2nd Ed., Prentice-Hall India (EEE), 1986.
4. J. Uffenberk, “Microcomputers and microprocessors”, 3rd Ed., Pearson Education, Asia (LPE), 2002.
5. C. Gilmore, “Microprocessors Principles and Applications”, 2nd Ed., McGraw-Hill International,1995.
6. D. Hall, “Microprocessors and Interfacing”, 2nd Ed., Tata-McGraw-Hill, 1999.
7. Treibel and Singh, “The 8088 and 8086 Microprocessors”, 4th Ed., Prentice-Hall India (EEE), 1991.
8. Mazidi, “The 8051 Microcontrollers & Embedded Systems”, Pearson Education Asia (LPE).
9. M. Predco, “Programming and Customizing the 8051 Microcontroller”, Tata McGraw-Hill EA., 1999.

Course Outcomes (COs):

The student will be able to:

1. Understand the basic architecture of 8-bit and 16-bit microprocessors with functional activities of different units.
2. Write assembly language programs for solving scientific problems.
3. Know the different steps for execution of an instruction and associated control signals.
4. Design system using memory chips and peripheral chips for 8 bit 8085 microprocessor.
5. Understand how to handle and process interrupts of the microprocessor.
6. Design system for communications between microprocessor and peripherals.

Course Articulation Matrix:

Microprocessors and Assembly Language Programming	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	2	2	1		1			2		1	2	2	3		1
CO2	2	1	2	3	1				1		1	2	3	2		1
CO3	2	3	2	2	1				1		1	2	2	3		1
CO4	2	2	3	2	1				1		2	2	1	3	1	2
CO5	2	3	1	1								2	2	3		2
CO6	2	2	3	2	1				1			2	2	3	1	2

Course code	CSE/PC/B/T/224
Category	Professional Core
Course title	**Data Communication**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – II
Pre-requisites (if any)	

Syllabus:

Introduction: Overview of Data Communications, Networks and Network models (OSI, TCP/IP), Protocols and standards [1L]

Data and signals: Analog and digital signals, Periodic and nonperiodic signals, Signal analysis, Composite signals, Time and Frequency domains, Bandwidth, Wave symmetry, Linear and non-linear mixing of signals. [2L]

Transmission Impairment: Attenuation, Distortion, Noise - correlated and uncorrelated noises and their categories, Harmonic distortion and intermodulation distortion, Data rate limits for noisy and noiseless channels [2L]

Performance: Bandwidth, Throughput, Latency, Bandwidth-Delay Product, Jitter [1L]

Digital Transmission: Problems with digital transmission, Different line coding schemes, Block coding schemes, Scrambling techniques; Analog to Digital Conversion – Sampling techniques, Sampling theorem, Pulse amplitude modulation, Pulse code modulation, Differential pulse code modulation, Delta modulation (along with advantages and disadvantages of each technique), Transmission modes (serial and parallel). [4L]

Analog Transmission: Concepts of carrier signal, modulating signal and modulated signal; Amplitude modulation – double sideband suppressed carrier, double sideband transmitted carrier, single sideband; Frequency modulation – Narrowband FM and wideband FM; Digital to analog conversion – Amplitude shift keying, Frequency shift keying, Phase shift keying, Quadrature amplitude modulation, Performance. [4L]

Transmission Media: Guided (wired) media – Twisted pair cable, Coaxial cable and Fibre optic cable, Construction, categories and connectors of each type, Performance, Advantages and disadvantages and applications of each type of media, Different propagation modes through fibre optic cable, Unguided (wireless) media – Different propagation modes, Radio waves, Terrestrial microwaves, Infrared, Applications and performances, Satellite communication. [4L]

Multiplexing and Spreading: Concept of multiplexing, Frequency division multiplexing, Time division multiplexing – Synchronous and Statistical time division multiplexing, Handling variable length data, Pulse stuffing, Concept of spreading spectrum, Frequency hopping spread spectrum and Direct sequence spread spectrum. [6L]

Modems and Interfaces: Dial-up modems, modem speed, standards; other modems; Interface standards. [4L]

Error Detection and Correction: Types of errors, Basic concepts of error detection and correction, Redundancy, Hamming distance, Error detection – Simple parity check codes, Two-dimensional parity check, Cyclic redundancy check, Polynomials and cyclic code analysis, Checksum, Error correction – Hamming code. [6L]

Protocols for Data Communication: Flow control and Error control, Stop and Wait protocol and its efficiency, Sliding window protocols - Go-back-N and Selective repeat, Piggybacking, HDLC, Point-to-point protocol. [6L]

Suggested Readings:
1. Data & Computer Communications, William Stallings, Pearson Education
2. Data Communications and Networking, Behrouz A Forouzan, McGraw Hill
3. Electronic Communications Systems, Tomasi, Pearson Education
4. Digital Communications, Haykin, Wiley

Course Outcomes (CO):
On completion of this course, students should be able to:

- CO1 Understand the fundamentals of network design, characteristics of analog and digital signals, relationship between data and signals, network topologies and devices and the concept of data communication within the network environment. (K2)
- CO2 Understand the basics of transmission medium and explain how impairments (noise, attenuation and distortion) affect signal traveling through a transmission medium (noiseless and noisy channel). (K2, K3)
- CO3 Understand and describe the concepts of digital transmission of analog and digital data, encoding techniques, conversion techniques used to convert digital data and analog signals to digital signals for parallel and serial transmission. (K2)
- CO4 Understand the describe the concepts of analog transmission of digital and analog data, methods, and the procedures involved in converting digital data and analog low-pass to band-pass analog signals. (K2)
- CO5 Understand and illustratively explain errors in communication, error detection and correction mechanisms. (K2, K4, K5)
- CO6 Explain the concepts of logical link control with reference to framing, flow and error control. (K2, K3, K4)

CO-PO mapping Course Articulation Matrix

CO **Data Communication**	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	3	2	2							1	2	1	3		
CO2	3	3	3	2	1						1	2	1	3		
CO3	3	2	2	2	1						1	2	1	3		
CO4	3	3	3	3	2	2	1	2	2	2	2	3	1	3		
CO5	3	3	3	2	2	1	1	2	2	2	2	3	3	2		
CO6	3	3	3	2	2	2	1	2	2	2	2	3	3	2		

Course code	CSE/PC/B/T/225
Category	Professional Core
Course title	**Graph Theory and Combinatorics**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – II
Pre-requisites (if any)	A basic knowledge of Discrete Mathematics and pre-college mathematics

This course covers the theory of graphs and networks for both directed and undirectedgraphs. Topics include graph isomorphism, Eulerian and Hamiltonian graphs, matching, covers, connectivity, coloring, and planarity. There is an emphasis on applications to real world problems and on graph algorithms such as those for spanning trees, shortest paths, and network flows.

Syllabus:

Introduction to Graph Theory [4L]
Definitions and Examples, Subgraphs, Complement of a graph, Graph Isomorphism, Degree, Directed and undirected graphs, weighted and unweighted graphs, dual graph.

Path, Cycles, Coloring [8L]
Walk, Trail, Path, Cycle, Euler Trails and Circuits, Planar Graphs, Hamilton Paths and Cycles, Vertex coloring, Edge coloring, Chromatic Polynomials.

Trees [4L]
Definitions, Properties and Examples, Rooted Trees, Trees and Sorting, Binary Trees, Weighted Trees and Prefix Codes
Graph Algorithms [7L]
Graph Traversals, Shortest Path Algorithms, Minimal Spanning Trees – the algorithms of Kruskal and Prim, Max-flow Min-cut Theorem, Matching.

Principle of Inclusion and Exclusion [6L]
The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials.

Generating Functions [6L]
Definition and Examples – Calculational Techniques, Partitions of Integers, The Exponential Generating Function, The Summation Operator.

Recurrence Relations [5L]
First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients, The Non-homogeneous Recurrence Relation, Solving Recurrences by Generating Functions, Fibonacci Numbers and Golden Ratio.

References:
1. F. Harary: Graph Theory
2. N. Deo: Graph Theory with Applications to Engineering and Computer Science 3. A. Tucker: Applied Combinatorics

Course Outcomes (COs):

CO 1. To create (K6) knowledge for analyzing (K4) problems and interpret/explain (K6) them with the acquired knowledge develop (K5) solutions through a mathematical framework that supports engineering, science, and mathematics.
CO 2. To characterize (A5) given problems by analyzing their arguments, in relation to their premises, assumptions, contexts, and conclusions and associating (K2) them with the mathematical designs (S5) used to express physical and natural laws, followed by computing (K3)
CO 3. Problem solving (S4) is naturalized (S5) through critical and analytical thinking.
CO 4. Be conversant with the mathematical precision and rigor that is necessary for computer science and be able to present (A2) it with accomplished (S3) arguments.

Course Articulation matrix

Graph Theory and Combinatorics	Program Outcomes									Program Specific Outcomes						
CO	PO 1	PO 2	P O 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO 1	3	3	2	3		2					2	3	3	2		1
CO 2	3		3		2				2			3	3	2		
CO 3		3	2			3	2			3		3			2	1
CO 4	3	3		2	3					3		3	3	2	1	

Course code	CSE/PC/B/T/226
Category	Professional Core
Course title	**Computer Architecture**
Scheme and Credits	L–T–P: 3-0-0; Credits: 3.0; Semester – II
Pre-requisites (if any)	

Syllabus:

1. Introduction: Design objectives of a computer architect; cost and performance measures; benchmark & metrics; instruction set architecture classification; instruction format and semantics; memory addressing modes; instruction encoding principles; role of compilers; formal description of architecture; VHDL; AADL.

[2L]

2. Instruction level parallelism:
Basic principles of pipelines; structural, control and data hazards; instruction pipelines; branch prediction; pipeline scheduling and collision avoidance; optimizing pipeline performance; RISC & CISC pipeline examples. VLIW architecture; overview of proposed and commercial VLIW Systems.

[**2L**]

Superscalar architecture; basic objectives of superscalar processing; superscalar instruction issues; issue policies; instruction pairing rules; shelving; register renaming; load/ store reordering; the reorder buffer; instruction pipeline – D1, D2 execution and write-back stages; branch handling – delayed branch, multiway branch; case study – Power PC620, Pentium Pro. [**4L**]

Code scheduling for ILP processor; basic block scheduling; loop scheduling; global scheduling. [**2L**]

Data parallel architecture: Basic idea of data parallelism; connectivity – nearest neighbour, tree, pyramid, mesh, hypercube and reconfigurable networks; different classes of data parallel architecture – SIMD, associative, neural, data parallel pipeline, systolic and vector architectures. [**3L**]

SIMD architecture; features – granularity, connectivity, processor complexity & local autonomy; fine grained SIMD overview; an example – the Massively Parallel Processor; coarse grained SIMD overview; an example – the CM5; SIMD algorithm examples – matrix multiplication/ inversion, sorting/ searching. [**3L**]

Systolic architecture; introduction; systolic design space; comparison with multidimensional pipeline; spatial convolutions; case study – the WARP processor. [**2L**]

Vector architecture; principles of vectorization; pipelined & parallel stream implementation of vector machine; case study – the CRAY-1, C-90 and the Convex C4/X4 system. [**2L**]

Thread/ Process level parallelism: Basic architectural concepts; scalable parallel architecture; design issues for scalable MIMD computers. [**1L**]

Multi-thread implementation on sequential control flow model; case study – the Dencolor HEP machine, the MIT ‘Sparcle’ machine. [**3L**]

Dataflow architecture; the classical static dataflow machine proposed by J Dennis; tagged token dataflow machine; explicit token-store architecture; dataflow model verification using simple/ coloured Petri Net. [**3L**]

Shared memory MIMD architecture; systems using single & multiple shared buses; blocking & non-blocking interconnection networks such as cross-bar and other MINs. [**2L**]

Cache coherence problem; hardware & software coherence policies – write-invalidate, write- update, write-through and write-back policies; snoopy protocol. [**2L**]

Synchronization; spin-lock; event ordering in coherent systems. [**1L**]

Uniform memory access (UMA) machine example; non-uniform memory access (NUMA) machine example; cache coherent NUMA (CCNUMA) machine example; case study – the SUN Enterprise 6000. [**2L**]

RISC Architecture: comparison between CISC & RISC concepts; RISC machine, features; hardwired control; horizontal machine code format; register file; jumps & delay slots. [**2L**]

Special Architecture: Architectural considerations for low power hand held mobile devices, embedded systems. [**1L**]

Parallelization: Parallel program development environment and software tools; mapping application onto multi-computers. [**1L**]

Performance Evaluation: Role of performance; performance metrics; Amdahl’s law; benchmarks; the SPEC benchmarks; SPEC95 for Pentium & Pentium Pro; SPEC 2000 benchmarks; MIPS as performance metric; native, peak & relative MIPS & FLOPs as performance measure; synthetic benchmarks; price-performance metric. [**2L**]

Suggested Readings:

1. Computer Architecture: A Quantitative Approach by J. L. Hennessy and D. A. Patterson, 3^{rd} & 4^{th} ed, Elsevier.
2. Advanced Computer Architecture: Parallelism, Scalability, Programmability by Kai Hwang, TMH.
3. Computer Organization and Design: The Hardware/ Software Interface, by D. A. Patterson and J. L. Hennessy, 4^{th} ed, Morgan Kaufmann.
4. Advanced Computer Architecture: A Design Space Approach, by D. Sima, T. Fountain and P. Kacsuk, Pearson Education.
5. Advanced Microprocessors by D. Tabak, McGraw-Hill.
6. The Pentium Microprocessor by J. L. Antonakos, Prentice Hall.

Course Outcomes (CO):

At the end of this course, each student should be able to:

CO1: Understand and analyse the issues involved in pipelining.

CO2: Recognize and analyse the application of pipelining to vector supercomputers for number-crunching applications.

CO3: Develop familiarity with MIMD architectures and understand the consistency problems involved.

CO4: Understand and apply benchmarks and performance metrics

CO5: Recognize the need to develop newer approaches in view of the ever-increasing and diverse needs for computation.

CO6: Understand the interplay of different factors like hardware limitations, efficiency of software, energy efficiency, end-user expectations, etc. in the development of architectures.

Course Articulation matrix:

Computer Architecture	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	3	1	2	1							3	2	3		
CO2	3	3	1									1	3	2	3	
CO3	3	3	2	3	2							3	3	3		
CO4	3	3		3	3	1					1	2	2	1		
CO5						2	3		1	2		3			3	2
CO6	3		3	3	2		2			2			3	3	2	1

Course code	CSE/PC/B/S/221
Category	Professional Core
Course title	**Hardware Design Lab**
Scheme and Credits	L–T–P: 0-0-2; Credits: 1.5; Semester – I
Pre-requisites (if any)	

Syllabus:

Experiment 1: Implement a variable frequency system clock with 555

Experiment 2: Implement a digital lock.

Experiment 3: Implement an automatic counter of the number of people entering an Auditorium.

Experiment 4: Implement a traffic light controller system for 4-way crossing.

Experiment 5: Implement a temperature controller system using a look up table stored in an EPROM.

Experiment 6: Design a 4-bit ALU.

Course Outcomes (COs):

At the end of the course a student shall be able to:

CO1: Design a variable frequency system clock

CO2: Design some real life application using counters.

CO3: Design some basic application using EPROM.

CO4: Design an ALU.

Hardware Design Lab	PO1	PO 2	PO 3	PO 4	P O 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	2	2	3											3		
CO2	2	2	3											3		
CO3	2	2	3											3		
CO4	2	2	3											3		

Course code	CSE/PC/B/S/222
Category	Professional Core
Course title	**Object Oriented Programming Lab**
Scheme and Credits	L–T–P: 0-0-2; Credits: 1.5; Semester – I
Pre-requisites (if any)	

Syllabus:

1. Problem on the use arrays, String structure (using Java and Python)
2. Problems on basic object oriented features like encapsulation, overloading, overriding, inheritance, abstract class, interfaces (using Java and Python)

3. Handling concurrency and exceptions (using Java and Python)
4. Exploring APIs for different data structures (using Java and Python)
5. Designing GUI applications (using Java and/or Python)
6. Problems on object oriented design of systems and implementation of the same (using Java and/or Python)

Course Outcomes

CO1: Understand and implement fundamental and OOP features through Java and Python Programming
CO2: Applying OOP concepts in solving problems and demonstration with ethical practice
CO3: Understand and utilize collection classes in Java and string classes in Python.
CO4: Develop applications in Java and Python

		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Object Oriented Programming Lab	CO1	2	3	3	2	2			1	1		1	3
	CO2	2	3	3	3	2			2	1		1	3
	CO3	2	3	3	3	2			2	2		2	2
	CO4	2	2	3	3	2			2	2		3	2

		PSO1	PSO2	PSO3	PSO4
Object Oriented Programming Lab	CO1	3		1	2
	CO2	3		1	2
	CO3	3		1	2
	CO4	3		1	2

Course code	CSE/PC/B/S/223
Category	Professional Core
Course title	**Microprocessor Lab**
Scheme and Credits	L–T–P: 0-0-2; Credits: 1.5; Semester – I
Pre-requisites (if any)	

Syllabus:

It covers different aspects of writing and executing assembly language programs. In particular, writing assembly language programs for performing different operations on data, such as addition, subtraction, multiplication of numbers and other operations using different instructions. Finding smallest and largest numbers in an array. Sorting an array. Communications between microprocessor and peripherals.

Course Outcomes (COs):

The student will be able to

1. Understand the basic architecture of 8-bit and 16-bit microprocessors with functional activities of different units.
2. Write assembly language programs for solving scientific problems.
3. Know the different steps for execution of an instruction and associated control signals.
4. Design system and write the associated programs for communications between microprocessor and peripherals.

CO **Microprocessor Lab**		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
1		3	2	2	1		1			2		1	2
2		2	1	2	3	1				1		1	2
3		2	3	2	2	1				1		1	2
4		2	2	3	2	1				1			2

CO **Microprocessor Lab**		PSO1	PSO2	PSO3	PSO4
1		2	3		1
2		3	2		1
3		2	3		1
4		2	3	1	2