

Department of Computer Science & Engineering
Revised Course structure of B. Tech, CSE (session 2024-25)

3 rd SEMESTER							
SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs/WEEK	CREDIT
THEORY							
01	BS(CS)307	Mathematics- III	3	0	0	3	3
02	PC(CS)301	Computer Organization	3	1	0	4	4
03	PC(CS)302	Data structure & Algorithms	3	1	0	4	4
04	PC(CS)303	Communication Engineering	3	0	0	3	3
05	HS(CS)305	Economics for Engineers	2	0	0	2	2
06	MC(CS)301	Environmental Sciences	2	0	0	2	2
SESSIONAL/PRACTICAL							
01	PCL(CS)304	Computer Organization Lab	0	0	3	3	1.5
02	PCL(CS)305	Data structure Lab	0	0	3	3	1.5
03	PCL(CS)306	IT Workshop (Python)	0	0	3	3	1.5
04	PCL(CS)307	Communication Engineering Lab.	0	0	3	3	1.5
05	TRN-SSD(CS)301	Soft Skill Development (intra institutional)	0	0	0	0	1
TOTAL			16	2	12	30	25

Name of the course		Mathematics-III	
Course Code: BS(CS) 307		Semester: 3rd	
Duration: 6 month		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 3hrs/week		Mid Term 1 Exam: 15 Marks	
Tutorial: 0hrs/week		Mid Term 2 Exam: 15 Marks	
Practical: 0 hrs/week		Other Assessment tools (Assignment, Quiz etc.): 20 Marks	
Credit Points: 3		End Semester Exam: 100 Marks (50% weightage for final reckoning i.e., 50 mark)	
Objective:			
1.	To understand the use of periodic signals and Fourier series to analyze circuits.		
2.	To understand gradient, divergence and curl using the calculus and multiple variable.		
3.	To understand Green, Gauss and stokes theorem using integral of a function.		
4.	To learn analytical technique for finding solution of higher order differential equation.		
5.	To create mathematical models using first order differential equation.		
6.	To understand basic concept of graph theory.		
Pre-Requisite			
1.	Mathematics –I (BS(CS/IT)101		
2.	Engineering Mathematics (UG level)		
Unit	Content	Hrs	Marks
1	Module 1: Fourier series & Fourier transforms Introduction to infinite series, convergence and divergence. Periodic functions, Dirichlet's condition. Fourier series of periodic functions with period 2 and arbitrary period. Half range Fourier series. Fourier transforms properties, Sine & Co-Sine Fourier transforms, convolution, inverses and Parseval's Identity.	8	
2	Module 2: Multivariable Calculus (Differentiation) Limit, continuity and partial derivatives, Chain rule, Implicit function, Jacobian, Directional derivatives, Total derivative; Maxima, minima and saddle points; Gradient, curl and divergence and related problems.	7	
3	Module 3: Multivariable Calculus (Integration) Double and triple integrals (Cartesian and polar), change of order of integration in double integrals, Change of variables (Cartesian to polar). Theorems of Green, Gauss and Stokes (Statement only) and related problems.	8	
4	Module 4: Ordinary Differential Equation First Order Differential Equation, Exact, Linear and Bernoulli's equations, Equations of first order but not of first degree: equations solvable for p, equations solvable for y,	7	

	equations solvable for x and Clairaut's form, general & singular solution. Second order linear differential equations with constant coefficients, D-operator method, method of variation of parameters, Cauchy-Euler equation.		
5	Module 5: Graph Theory Basic Concept of graph, Walk, Path Circuit, Euler and Hamiltonian graph, diagraph. Matrix Representation: Incidence & Adjacency matrix. Tree: Basic Concept of tree, Binary tree, Spanning Tree, Kruskal and Prim's algorithm for finding the minimal spanning tree.	6	

Course outcomes

After completion of the course, a student would be able to:

CO 1	Analyse circuits using periodic signals and Fourier series.
CO 2	Understand the concept of limits, continuity and differentiability of functions of several variables. Analytical definition of partial derivative. Maxima and minima of functions of several variables Define gradient, divergence and curl of scalar and vector functions.
CO 3	evaluate multiple integrals and apply the techniques to different physical problems.
CO 4	solve first and second order ordinary differential equations by applying different techniques and also will be able to formulate differential equations for model systems and problems of engineering sciences.
CO 5	apply the basic concepts of graph theory to network analysis, data analytics and many other branches of computer science.

Learning Resources:

1.	S.K Mapa, Real Analysis, Sarat
2.	Charles H.C. Little, Kee L. Teo, Bruce van Brunt, Real analysis via sequence and series, Springer
3.	Douglas Brent West, Introduction to Graph Theory, Prentice Hall.
4.	Robert Wrede, Murray Spiegel, Schaum's Outline of Advanced Calculus, Third Edition, Schaum's outline
5.	S.L. Ross, Differential equation, Willey.
6.	Clark John, Holton Derek Allan, A First Look at Graph Theory, World Scientific.

Name of the course		Computer Organization	
Course Code: PC(CS)301		Semester: 3rd	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 3 hrs/week		Mid Semester 1 Exam: 15 Marks	
Tutorial: 1 hrs/week		Mid Semester 2 Exam: 15 Marks	
Practical: 0 hrs/week		Assignment, Quiz, Attendance: 20 Marks	
Credit Points: 4		End Semester Exam: 100 Marks (100 marks converted to 50)	
Objective:			
1	To understand the structure, function, and design of computers.		
2	To study data representation, memory systems, different processor architecture and their performance measurement parameters.		
3	To develop the concept of instruction set of a processor and control unit design.		
4	To design pipeline processor architecture.		
Pre-Requisite			
1.	Basic knowledge of Digital Logic.		
Unit	Content	Hrs	Marks
1	Introduction: History of computing, von Neumann machine, Instruction and data, fixed-point and floating-point numbers, errors, IEEE standards	4	
2	Processor design: Instruction Set Architecture-Instruction format, opcode optimization; operand addressing; Instruction implementation-data movement, branch control, logical, Input/output and debugging instructions; arithmetic instruction implementation-addition and subtraction, multiplication-division, 2's complement multiplication; Booth's algorithm-theory and examples; bit-pair algorithm; high performance arithmetic; Pipelining: Basic concepts, hazards, and performance	10	
3	Control unit design: Hardwired control, micro-programmed control design – micro-instruction formats, control optimization;	6	
4	Memory subsystem: Registers, Memory technology, memory interfacing, Memory hierarchy-introduction to virtual memory system; cache memory – performance, address mapping, content addressable memory (CAM)	9	
5	Peripherals: Basic properties, bus architectures, interfacing of I/O devices, data transfer schemes –programmed I/O, DMA, mass storage, RAID	7	
Course Outcomes:			
After completion of this course students will be able to-			
CO1	Explain the evolution of computer systems and demonstrate understanding of number systems, data representation, and IEEE floating-point standards.		

CO2	Design and analyze instruction sets, addressing modes, and arithmetic operations including Booth's and bit-pair multiplication algorithms.
CO3	Differentiate between hardwired and microprogrammed control units and apply microinstruction formats for control unit design.
CO4	Evaluate memory hierarchy and technologies including cache memory, virtual memory, and CAM, and their impact on system performance.
CO5	Analyze pipelined and superscalar processor architectures and assess their impact on instruction throughput and processor performance.
Learning Resources:	
1	Mano, M.M., "Computer System Architecture", PHI
2	Behrooz Parnami "Computer Architecture", Oxford University Press
3	Hayes J. P., Computer Architecture & Organization, McGraw Hill
4	Hamacher, Computer Organization, McGraw Hill,
5	N. Senthil Kumar, M. Saravanan, S. Jeevananthan, Microprocessors and Microcontrollers OUP
6	Chaudhuri P. Pal, Computer Organization & Design, PHI
7	P N Basu- Computer Organization & Architecture, Vikas Pub

Name of the course		Data structure & Algorithm	
Course Code: PC(CS)302		Semester: 3rd	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 3 hrs/week		Mid Semester 1 Exam: 15 Marks	
Tutorial: 1 hrs/week		Mid Semester 2 Exam: 15 Marks	
Practical: 0 hrs/week		Assignment, Quiz, Attendance: 20 Marks	
Credit Points: 4		End Semester Exam: 100 Marks (100 marks converted to 50)	
Objective:			
1.	Introduce fundamental data structures like arrays, linked lists, stacks, queues, trees, and graphs, and demonstrate their use in software development.		
2.	Develop students' ability to analyze algorithm efficiency using time and space complexity and asymptotic notations.		
3.	Enable design and implementation of fundamental operations on various data structures.		
4.	Develop problem-solving skills using appropriate data structures for tasks such as searching, sorting, and traversals.		
5.	Lay the groundwork for further study in algorithms, software design, and systems development.		
Pre-Requisite			
1.	Programming for problem solving (ES(CS/IT) 204)		
Unit	Content	Hrs	Marks
1	Module 1: Introduction to Algorithms and Complexity Definition and characteristics of algorithms, Algorithm specification, pseudocode, Time and space complexity, Asymptotic notations (O , Ω , Θ), Recurrence relations (basic introduction)	4	
2	Module 2: Linear Data Structures Arrays and Strings: One-dimensional and multi-dimensional arrays, Operations and complexity analysis, String representation and manipulation Linked Lists: Singly, doubly, and circular linked lists, Operations: insert, delete, search, traverse, Memory management and dynamic allocation Stacks and Queues:	10	

	<p>Stack operations and applications: postfix evaluation, recursion</p> <p>Queue types: simple, circular, priority queues</p> <p>Implementations using arrays and linked lists</p>		
3	<p>Module 3: Non-Linear Data Structures</p> <p>Trees: Tree terminology and binary trees, Binary Search Trees (BST), Traversal methods-in-order, pre-order, post-order (recursive and iterative) BST operations- insertion, deletion, search Introduction to balanced trees (concept of AVL tree only)</p> <p>Heaps: Min-heap and max-heap, Heap operations and applications, Heap sort</p> <p>Graphs: Graph terminology and representations (adjacency matrix/list), Breadth-First Search (BFS) and Depth-First Search (DFS)</p>	10	
4	<p>Module 4: Searching and Sorting Algorithms</p> <p>Searching: Linear search, Binary search, Analysis of efficiency</p> <p>Sorting: Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort, Heap sort, Time and space complexity analysis and comparisons</p>	8	
5	<p>Module 5: Hashing and File Structures</p> <p>Hash functions and collision resolution (chaining, open addressing)</p> <p>Introduction to file organization and indexing</p>	4	
Course outcomes			
After completion of the course, a student would be able to:			
CO 1	Explain the concepts of algorithms, time and space complexity, asymptotic notations, and recurrence relations.		
CO 2	Implement linear data structures such as arrays, linked lists, stacks, and queues to perform basic operations.		
CO 3	Construct non-linear data structures including trees, heaps, and graphs using appropriate representations.		
CO 4	analyse the performance of searching and sorting algorithms using time and space complexity.		

CO 5	Demonstrate the use of hashing and basic file organization techniques for efficient data access.
Learning Resources:	
1.	Horowitz, Sahni, Anderson-Freed: Fundamentals of Data Structures in C (Second Edition), Universities Press, 2008.
2.	Lipschutz, Seymour. Data Structures (Schaum's Outline Series). Tata McGraw-Hill Publishing Company Ltd., New Delhi.
3.	T.H. Carmen, C.E. Leiser son, R. Rivest and C. Stein: Introduction to Algorithms, (Second/Third Edition), PHI, 2009.
4.	Aho, Alfred V., Hopcroft, John E., and Ullman, Jeffrey D. Data Structures and Algorithms. Reading, MA: Addison-Wesley, 1983. ISBN: 978-0201000238.
5.	R. Sedgewick: Algorithms in C, Pearson, 2004.
6.	Steven S Skeena, Algorithm design manual, 2nd Edition, Springer.
7.	Steven S Skeena, Miguel A. Revilla, Programming Challenges: The Programming Contest Training Manual (Texts in Computer Science) Springer.

Name of the course:	Communication Engineering
Course Code: PC(CS)303	Semester: 3rd
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory Contact Hrs.: 3 hrs. /week	Mid Semester-1 Exam: 15 Marks
Tutorial Contact Hrs.:	Mid Semester-2 Exam: 15 Marks
Credit Point: 3	Assignment, Quiz & class Attendance: 20 Marks
	End Semester Exam: 100 Marks (to be mapped into 50 marks)
Objective:	
1.	To study Amplitude Modulation and Frequency Modulation techniques

2.	To study pulse modulation techniques and line codes.
3.	To study different shift keying techniques
4.	To study different aspects of satellite communication

Pre-Requisite:

1.	Mathematics I BS(CS/IT) 101, Mathematics II BS(CS/IT) 205, Digital Electronics [ES(CS/IT)307
----	--

Unit	Content	Hours	Marks
1.	Introduction to Communication Engineering, need of Modulation, Amplitude Modulation (AM): Concept of AM, Calculation of Modulation Index, total transmitted power of AM, DSB-SC modulation & SSB-SC modulation techniques, calculation of Bandwidth and Savings of power, Demodulation of AM, Superheterodyne Receiver	12	
2.	Frequency Modulation (FM): Concept of FM, Direct & Indirect Method, Bandwidth calculation of FM, Demodulation of FM. Phase Modulation (PM): Concept of PM, generation of PM from FM.	05	
3.	Pulse & Digital Communication: Sampling Theorem, aliasing effect, natural and flat top sampling, PAM, PWM, PPM, basic concept of Pulse Code Modulation (PCM), concept of quantization and quantization error, Compounding, DPCM, Delta Modulation and Adaptive Delta Modulation, signal to quantization noise ratio in PCM system. ASK, FSK, PSK, QPSK	12	
4.	Data Formatting: NRZ-Unipolar, NRZ-polar, NRZ-Bipolar, RZ-Bipolar, Manchester Coding, Synchronous and Asynchronous Data Transmission, Concept of Satellite Communication	07	

Course Outcomes:

After completion of this course the students will be able to -

CO1	Explain the necessity of Modulation and how to transfer information from one place to another place using Amplitude Modulation, Frequency Modulation and Phase Modulation.
CO2	Apply the concept of sampling and quantization for analog to digital signal conversion.
CO3	Compare various techniques of digital communication techniques.
CO4	Compare different line coding techniques.
CO5	Compare Satellite Communication system with terrestrial communication system.

Learning Resources:

1.	Modern Digital and Analog Communication Systems by B.P. Lathi, Published by Oxford University Press.
2.	An Introduction to Analog and Digital Communications by Simon Hay kin (Wiley India)
3.	Principles of Communication Engineering by Taub H. & Shilling D.L.- TMH

4.	Introduction to Digital and Data Communication – Michael A. Miller, Jaico Publishing House
5.	Communication Systems by A. B. Carlson, Published by McGraw-Hil
6.	Principles of Analog and Digital Communication by Jerry D Gibson, Published by MacMillan.
7.	A Text Book of Analog and Digital Communication by A Kumar, Umesh Publication
8.	Communication Systems (Analog and Digital) by Sanjay Sharma, Published by S.K. Kataria& Sons
9.	Modern Electronic Communication, Principles and Practice- Sharma &Sinha, Dhan Patrai Publishing Company (p) Ltd

Name of the course		Economics for engineers	
Course Code: HS(CS)305		Semester: 3rd	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 2 hrs./week		Mid Term Exam I: 15 Marks	
Tutorial: Nil		Mid Term Exam II: 15 Marks	
Practical: Nil		Assignment.: 20 Marks	
Credit Points: 2		Semester End Exam: 100 Marks (Two third weightage for final reckoning i.e., 50 marks)	
Objective:			
1.	To familiarize with the analysis of different economic alternatives using appropriate concepts and methods of Economics for Engineers.		
Pre-Requisite			
1.	Class 12th standard knowledge of Mathematics.		
Unit	Content	Hours	Marks.
1	<ul style="list-style-type: none"> Introduction to Economics for Engineers – Basic Introduction to Economics, Productive resources, Scarcity and the Economic Problem, Efficiency, Production possibility frontier, Demand and Supply, Circular flow of income, Forms of Business Organization, Financing business needs, Basic Financial Statements and ratios, Engineering & Economics, Definition and Scope of Economics for Engineers, Role of Engineers in Economic Decision making, Decision-Making Process. Engineering Cost Concepts and Cost Estimation Techniques – Fixed, Variable, Marginal & Average costs, Semi-variable and Step cost, Product and Period cost, Direct and Indirect cost, Sunk cost, Shutdown cost, Opportunity cost, Recurring and Nonrecurring costs, Anticipated and Unanticipated costs, Differential or Incremental costs, Cash cost vs. Book 	7	

	<p>costs, Explicit and Implicit costs, Out of pocket costs and Imputed costs, Life-Cycle Costing. Approaches to cost estimation, Types of Estimate, Cost Estimation Models - Per Unit Model, Segmenting Model, Cost Index Model, Power-Sizing Model, Learning Curve Model, Benefits and difficulties in estimation.</p> <ul style="list-style-type: none"> • Break-even analysis- Basic concept, terminology and assumptions, Derivation of break-even point, Profit Volume (P/V) ratio, Margin of Safety, Uses and limitations of break-even analysis. 		
2	<ul style="list-style-type: none"> • Cash Flow, Interest and Equivalence: Cash Flow – Diagrams and Cash Flow Statement, Time Value of Money, Real, Nominal & Effective Interest rate. Different Interest Formulae and their application. 	3	
3	<ul style="list-style-type: none"> • Capital budgeting and Project selection – Basic concept of capital budgeting, Net Present Value (NPV) Analysis, Internal Rate of Return (IRR) Analysis, Incremental analysis, Future Worth Analysis, Payback period, Evaluation of Public Projects and Benefit-Cost Ratio Analysis, Sensitivity analysis 	6	
4	<ul style="list-style-type: none"> • Uncertainty in Future Events - Uncertainty and Risk, Types of risk, Risk vs. Return, Application of Probability to analyse risk, Using Expected Value, Variance, and Coefficient of Variation to measure return and risk; Economic Decision Trees. • Inflation and Price Change – Definition, Stages, Effects, Causes, Price Change with Indexes, Types of Indexes, Composite vs Commodity Indexes, Tests for Index numbers, Use of Price Indexes in Engineering Economic Analysis. 	5	
5	<ul style="list-style-type: none"> • Depreciation and Replacement Analysis - Basic aspects of depreciation, Depreciation, depletion and amortization, Various methods of calculating depreciation; Replacement analysis – Basic aspects, Types of maintenance, Replacement Analysis Decision Map, Minimum Cost Life of a New Asset. 	3	

Course outcomes

After completion of the course, a student would be able to:

CO 1	Analyse short term alternatives using basic principles of Economics, Cost estimation and break-even analysis.
CO 2	Examine the concept of Interest and Equivalence.
CO 3	Inspect the feasibility of projects using various methods like Net Present Worth, Future Worth, Internal rate of Return and Benefit Cost Analysis.
CO 4	Identify the effects of uncertainty and price change.
CO 5	Apply Depreciation and Replacement Analysis using appropriate methods in relevant problems.

Learning Resources:

1.	R. Panneerselvam: Engineering Economics, PHI.
2.	H.L. Bhatia & S.N. Maheswari: Economics for Engineers, second edition, Vikas Publishing House Pvt Ltd.
3.	Donald Newnan, Ted Eschenbach, Jerome Lavelle: Engineering Economics Analysis, OUP
4.	Sullivan and Wicks: Engineering Economy, Pearson
5.	Partha Chatterjee: Economics for Engineers, Vrinda Publications.
6.	James L. Riggs, David D. Bedworth, Sabah U. Randhawa: Economics for Engineers 4e, Tata McGraw-Hill.
7.	Dr. Shantanu Chakraborty, Nilanjan Singha Roy: Economics for Engineers, Law point Publications.
8.	N.G. Das, Statistical Methods, Tata McGraw Hill
9.	Chan S Park, Contemporary Engineering Economics, Parson

Name of the course	ENVIRONMENTAL SCIENCES
Course Code:	Semester: 3rd
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory: 2hrs/week	Mid Semester-1 Exam: 15 Marks
Tutorial: NIL	Mid Semester-2 Exam: 15 Marks
Practical: NIL	Assignment, Quiz & class attendance: 20 Marks
Credit Points: 2	End Semester Exam: 75 Marks (Two third weightage for final reckoning i.e., 50 marks)
Objective:	
1.	To provide knowledge about green computing and green strategies
2.	To learn about the various types of E waste and their composition, their harmful effects on the environment.
3.	To know about carbon footprint, quantification and measurement
4.	To get the idea about urban problems related to energy and to know about sustainable transportation.
5.	To learn various green technologies and their implementation's.

Pre-Requisite			
Class 12 standard knowledge of physics, chemistry, biology, mathematics			
Unit	Content	Hrs	Marks
1	Green computing: Green IT Fundamentals, business, IT and the environment, Green IT strategies; Drivers, Dimensions and Goals	4	
2	Green Assets and Modelling; Buildings, Data Centres, Networks and Devices, green enterprise architecture, Environmental intelligence –green supply chains	4	
3	Carbon footprint: definition, quantification, measurement, direct and indirect emissions, types, difference from Ecological Footprint.	3	
4	E waste: definition, global scenario, growth of Electrical and Electronics industry in India E waste generation in India, environmental and health impact of E waste	5	
5	Social issues and the Environment (a) Urban problems related to energy (b) Water conservation, rain water harvesting, water shed management (c) Resettlement and rehabilitation of people; its problems and concerns, (d) Climate change, global warming, acid rain, ozone layer depletion, nuclear	4	
6	Green technologies; a) renewable energy (solar, wind etc.), energy efficiency b) waste management (recycling, composting) c) sustainable transportation (electric vehicles, biofuels) d) green building design and pollution control e) green chemistry (carbon capture and storage) f) sustainable agriculture	4	
Course Outcome:			

After completion of the course the students will be able to-	
CO 1	Understand the environmental consequences of information technology and the importance of green computing
CO 2	Understand the components environmental and health impact of e waste
CO 3	Apply the strategies of e-waste management in real world scenarios.
CO 4	Analyse the environmental consequences of various technologies and industrial processes.
CO 5	Apply their knowledge in implementing policies that incentivize sustainable practices, regulate emissions and promote carbon offsetting initiatives.
CO6	apply their knowledge of various environment protection acts, “Environment Impact Assessment” (EIA) as and when required in setting up of new industries as well as expansion of industries in which they will be employed
Learning Resources:	
1	Anubha Kaushik, C.P. Kaushik, Perspectives in environmental studies, New Age International (P) Ltd, Publishers
2	Erach Bharucha, Textbook for Environmental Studies, University Grants Commission
3	D. D. Mishra, Fundamental concepts in Environmental Studies, S Chand & Co Ltd
4	Anil Kumar De, Arnab Kumar De, Environment and Ecology, New age international (P) Limited, Publishers
5	Environmental Chemistry by Anil Kumar De, Wiley Eastern Limited
6	Linda D. Williams, Environmental Science demystified, McGraw-Hill
7	Shashi Chawla, A Textbook of Environmental Studies, Tata McGraw Hill Education Private Limited.

Name of the course	Computer Organization Lab.
Course Code: PCL(CS)304	Semester: 3rd
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory: Nil	Attendance: 10
Tutorial: Nil	Preparation of Lab Report: 30
Practical: 3 hrs./week	Experimental data/ Precision of work done: 30
Credit Points: 1.5	Presentation/ analysis of the result: 10
	Viva Voce: 20

Pre-requisite:

1.	Digital Electronics laboratory (ESL(CS/IT)308)		
Module	Content	Hours	Marks
1.	Truth table verification of Universal gate. Design and implementation of basic gate using Universal gate.	03	
2.	Design and implementation of binary adder using full adder.	06	
3.	Design and implementation of increment circuit using half adder.	03	
4.	Design and implementation of BCD adder using binary adder.	03	
5.	Design and implementation of adder-subtractor composite unit.	03	
6.	Design and implementation of carry look ahead (CLA) adder.	03	
7.	Design and implementation of 4:1 / 8:1 MUX using 2:1 MUX.	03	
8.	Design and implementation of ALU.	03	
9.	Design and implementation of synchronous / asynchronous counter.	03	

Course Outcomes:

After completion of this course the students will be able to -

CO1	Design and verify the functionality of basic and universal logic gates through truth tables and gate-level circuit implementations.
CO2	Design and simulate binary adders and subtractors.
CO3	Design multiplexers (MUX) of higher order using lower order MUX.
CO4	Design and implement an ALU.

CO5	Develop and compare synchronous and asynchronous counters.
Learning Resources:	
1	Mano, M.M., “Computer System Architecture”, PHI.
2	M. Lotia, Modern IC data and substitution Manual, PHI

Name of the course		Data Structure & Algorithm Lab	
Course Code: PCL(CS)305		Semester: 3rd	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: Theory		Attendance: 10	
Tutorial: Nil		Preparation of Lab Report: 20	
Practical: 3		Experimental data/ Precision of work done: 30	
Credit Points: 1.5		Presentation/ analysis of the result: 20	
		Viva Voce: 20	
Objective:			
1.	Hands-on Implementation: Develop practical skills in implementing data structures and algorithms.		
2.	Problem-Solving: Reinforce problem-solving abilities through the design and implementation of efficient algorithms.		
3.	Analysis of Algorithmic Efficiency: Understand and evaluate algorithmic complexity using both time and space metrics.		
4.	Algorithmic Applications: Gain experience with the application of different data structures to solve real-world problems.		
5.	Debugging and Optimization: Learn to debug, optimize, and test data structure implementations.		
Pre-Requisite			
1.	Basic Programming concept using C		
Module	Content	Hours	Marks.
1	<p>Introduction & Basic Programming Setup: Introduction to programming environment, I/O operations, and dynamic memory management. Introduce algorithm complexity.</p> <p>Array Operations: Implement operations such as insertion, deletion, traversal, and searching on arrays. Test performance with large data.</p> <p>String Operations: Implement basic string operations such as concatenation, comparison, searching, and substring extraction.</p>	6	

2	Singly Linked List: Implement singly linked list with operations such as insertion, deletion, and traversal.	3	
3	Doubly Linked List: Implement doubly linked list with operations such as insertion, deletion, and traversal.	3	
4	Stack Operations: Implement a stack using arrays and linked lists. Use stacks for applications like postfix expression evaluation.	3	
5	Queue Operations: Implement simple queue, circular queue, and priority queue. Perform enqueue and dequeue operations.	3	
6	Binary Tree Operations: Implement binary tree with insert, delete, and traversal operations (pre-order, in-order, post-order).	3	
7	Binary Search Tree (BST): Implement BST with operations like insert, delete, search, and traversal. Evaluate efficiency of each operation.	3	
8	Graph Representation and Traversal: Implement graph representation using adjacency matrix/list. Perform BFS and DFS traversal.	3	
9	Searching and Sorting Algorithms: Implement searching algorithms (linear, binary) and sorting algorithms (bubble, insertion, quick, merge). Heap Operations: Implement min-heap and max-heap. Perform heap sort and evaluate its time complexity.	3	

Course outcomes

After completion of the course, a student would be able to:

CO 1	Demonstrate proficiency in using basic programming construct, I/O operations, debugging, dynamic memory allocation, and algorithmic complexity analysis while maintaining good programming practices.
CO 2	Implement array and string operations including insertion, deletion, searching, and performance testing on large datasets.
CO 3	Develop linear data structures such as singly and doubly linked lists, stacks, and queues using appropriate data representations.
CO 4	Construct non-linear data structures such as binary trees, binary search trees, heaps, and graphs with relevant traversal and manipulation algorithms.
CO 5	Evaluate the efficiency of different searching and sorting algorithms.
CO 6	Analyse the performance of different searching and sorting algorithms in different scenarios.

Learning Resources (Books):

1.	Aho, Alfred V., Hopcroft, John E., and Ullman, Jeffrey D. Data Structures and Algorithms Addison-Wesley, 1983. ISBN-13: 978-0201000238
2.	Horowitz, Ellis, Sahni, Sartaj. Fundamentals of Data Structures in C Pearson, 2nd Edition, 2006. ISBN-13: 978-8177584724, ISBN-10: 8177584727
3.	Cormann, Thomas H., Leiser son, Charles E., Rivest, Ronald L., and Stein, Clifford, Introduction to Algorithms, MIT Press, 3rd Edition, 2009. ISBN-13: 978-0262033848
4.	S. Lipschutz, Data Structures, Schaum's Outline Series, McGraw-Hill, 2006. ISBN-13: 978-0070634136

Software and Tools

1	IDE/Compiler: Code: Blocks, Eclipse, or Visual Studio Code (C/C++/Java programming), GCC, Turbo C++, or Clang (for C/C++ programming)
Online Resources	
1	Geeks for Geeks: https://www.geeksforgeeks.org/data-structures/
2	Khan Academy: https://www.khanacademy.org/computing/computer-science/algorithms
3	Coursera (University of California, San Diego): https://www.coursera.org/specializations/data-structures-algorithms

Name of the course	IT Workshop (Python).
Course Code: PCL(CS)306	Semester: 3rd
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory: Nil	Attendance: 10
Tutorial: Nil	Preparation of Lab Report: 30
Practical: 3 hrs./week	Experimental data/ Precision of work done: 30
Credit Points: 1.5	Presentation/ analysis of the result: 10
	Viva Voce: 20

Objective:

1. To implement Python programs using core Python programming concepts and functions
2. To understand Object Oriented Python Programming technique

Pre-Requisite

1. Basic Programming concept

Module	Content	Hours	Marks.
1	Python Fundamentals Python Character Set, Python Tokens, Basic structure of Python Program, Variables and assignments, Multiple Assignments, Dynamic Typing, Input and Output in Python, Data Types and Operators, Control Structure, Sequence Statements, Selection Statements, range () function, Iterative Statements, Jump Statements	6	
2	Strings Accessing Values in Strings, Traversing a String, String Operators, Built-In String Methods	3	
3	Lists Creating a List, Accessing Lists, Difference between String and List, Traversing a List, List Operations,	4	
4	Tuples Tuple vs List, creating a Tuple, Accessing Tuples, traversing a Tuple, Comparing Tuples, Common Tuple Operators, Packing and Unpacking Tuples, Tuples Built-In Functions, Deleting a Tuple	3	
5	Dictionary	3	

	Creating a Dictionary, Properties of Dictionary Keys, traversing a Dictionary, Accessing Keys or Values Separately, Nested Dictionary, Adding Elements to Dictionary, Updating Elements in a Dictionary, Deleting Element from a Dictionary, Dictionary Built-In Methods		
6	Introduction to Python Modules Math Module, Random Module, Statistics Module.	3	
7	Functions Scope, Parameter passing, passing strings, Default parameters, Return values, Positional parameters	3	
8.	Object Oriented Programming (OOP) With Python Basics of OOP, Class and Objects, Inheritance, Types of Inheritance	5	

Course outcomes

After completion of the course, a student would be able to:

CO 1	Interpret the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements
CO 2	Express proficiency in the handling of strings and functions
CO 3	Apply object-oriented programming concepts
CO 4	Determine the methods to create and manipulate Python programs by utilizing lists, tuples and dictionaries

Learning Resources:

1.	https://www.anaconda.com
2.	2. Rakesh K. Yadav, Srinivas Aru Konda, Monu Singh, Tapasya Dinkar, Dileep Kumar Yadav, Zero to Mastery in
3.	Python Programming, Vayu Education of India, ISBN: 9789389769364
4.	3. Pooja Sharma, Programming in Python, BPB Publications, ISBN: 9789386551276

Name of the course		Communication Engineering Lab.	
Course Code: PCL(CS)307		Semester: 3rd	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme, Total Marks: 100	
Theory: Nil		Attendance: 10	
Tutorial: Nil		Preparation of Lab Report: 30	
Practical: 3 hrs./week		Experimental data/ Precision of work done: 30	
Credit Points: 1.5		Presentation/ analysis of the result: 10	
		Viva Voce: 20	
Module	Content	Hours	Marks
1.	Amplitude Modulation and Demodulation	04	
2.	Frequency modulation and Demodulation.	04	

3.	Generation and Detection of PAM	03	
4.	Generation and detection of PWM & PPM	06	
5.	Generation and detection of ASK	03	
6.	Generation and detection of FSK	04	
7.	Time Division Multiplexing & Demultiplexing	06	
Course Outcomes:			
After completion of this course the students will be able to -			
CO1	Compare the Amplitude modulated (AM) and Frequency modulated (FM) signals.		
CO2	Measure the modulation index of amplitude modulated and frequency modulated signals.		
CO3	Compare PAM, PWM and PPM signal.		
CO4	Compare ASK and FSK signals with AM and FM signals.		
CO5	Identify the multiplexed signals at the output of TDM system and the corresponding demultiplexed signals at the receiver end.		
Learning Resources:			
1	Octave online https://octave-online.net/ the open-source alternative for simulation of the above experiments		

Name of the course		Soft Skill	
Course Code: TRN-SSD(CS)301		Semester: 3rd	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: Theory		Attendance: 10	
Tutorial: Nil		Preparation of Lab Report: 20	
Practical: 3		Experimental data/ Precision of work done: 30	
Credit Points: 1.5		Presentation/ analysis of the result: 20	
		Viva Voce: 20	
Objective:			
1.	Development of intra and Interpersonal Skills of Students.		
2.	To enhance Personal and Professional Growth of Student.		
3.	Enable the students to develop their communicative competence.		
Pre-Requisite			
Module	Content	Hours	Marks.
1	Introduction and personality development What are soft skills Need for soft skills Time Management: Goal setting, effective time utilization. Self-awareness: SWOT ANALYSIS	4	
2	Communication and public speaking	5	

	Email etiquette, writing clear and concise messages. ➤ Group discussion Public Speaking: Overcoming stage fright, structuring speeches, and engaging an audience. Presentation Skills: Designing visually appealing slides, delivering impactful presentations Non-verbal Communication: Body language, facial expressions, and gestures		
3	Emotional Intelligence and Self-Confidence Understanding Emotional Intelligence: Managing emotions, empathy, and self-regulation. Developing Self-Confidence: Overcoming self-doubt, positive self-talk, and body language. Stress Management: Techniques to handle stress and maintain emotional well-being. Adaptability and Resilience: Dealing with change and challenges effectively. Problem solving and Critical thinking	6	
4	Interpersonal Skills and Professional Etiquette ➤ Teamwork and Collaboration: Effective participation in teams, conflict resolution , and cooperation. ➤ Networking Skills: Building and maintaining professional relationships. ➤ Business Etiquette: Proper behaviour in a professional setting, dressing appropriately.	4	
5.	Interview Skills: Preparing for interviews,	4	
6.	Pronunciation practice. Practical usage of English grammar.	3	
Course outcomes			
After completion of the course, a student would be able to:			
CO 1	Students will be able to differentiate between hard skills and soft skills. By the completion of this unit students will understand the importance of interpersonal skills in today's competitive world.		
CO 2	Students will know about their self-assessments and personality. Students will learn negotiation skills and also, they will learn the importance of social etiquette to make the world a nicer place with simple social etiquette gestures and good manners.		
CO 3	Students will learn how to communicate effectively through various advanced activities specifically designed to encourage students to play an active role for enhancing their knowledge and developing learning strategies.		

CO 4	Students will have learnt how to make decisions when in dilemma, how to deal with complex problems and stress in the present world. They will also have learnt how to be a team player & how to develop leadership skills.
CO 5	After the completion of this unit students' will have learnt about how to develop positive attitude, how to work on Emotional Intelligence. They will also have learnt about the strategies to set crystal clear goals, how to think independently and innovatively.
Learning Resources:	
1.	Anderson, P. (2017). Technical Communication (9th ed.). Cengage Learning.
2.	Seely, J. (2013). Oxford Guide to Effective Writing and Speaking: How to Communicate Clearly (3rd ed.). Oxford University Press.
3.	Bovee, C., & Thill, J. (2017). Business Communication Today (14th ed.). Pearson.
4.	L. (2004). Basic Business Communication: Skills For Empowering the Internet Generation (10th ed.). McGraw-Hill/Irwin

4th SEMESTER

SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs/WEEK	CREDIT
THEORY							
01	BS(CS)408	Discrete Mathematics	3	0	0	3	3
02	PC(CS)408	Design & Analysis of Algorithm	3	1	0	4	4
03	PC(CS)409	Formal Language and Automata Theory	3	1	0	4	4
04	PC(CS)410	Computer Architecture	3	1	0	4	4
05	PC(CS)411	Operating Systems	3	1	0	4	4
SESSIONAL/PRACTICAL							
01	PCL(CS)412	Algorithm Lab	0	0	3	3	1.5
02	PCL(CS)413	Programming Lab using C++	0	0	3	3	1.5
03	PCL(CS)414	Operating System Lab	0	0	3	3	1.5
04	PCL(CS)415	Computer Architecture Lab	0	0	3	3	1.5
05	PROJ(CS)401	Minor Project 1 (Literature Survey)	0	0	2	2	1
TOTAL			15	4	14	33	26

Name of the course		Discrete mathematics	
Course Code: BS(CS) 408		Semester: 4th	
Duration: 6 month		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 3hrs/week		Mid Term 1 Exam: 15 Marks	
Tutorial: 1hrs/week		Mid Term 2 Exam: 15 Marks	
Practical: 0 hrs/week		Other Assessment tools (Assignment, Quiz etc.): 20 Marks	
Credit Points: 3		End Semester Exam: 100 Marks (50% weightage for final reckoning i.e., 50 mark)	
Objective:			
1.	To learn the concept of division algorithm and integer modulo n.		
2.	To understand counting techniques and combinatorics in the context of discrete probability.		
3.	To learn recurrence relations and generating functions.		
4.	To learn a given logic sentence and can check it's validity.		
5.	To understand Algebraic structures and classify Boolean function.		
6.	To understand basic concept of graph theory, Dual and planar graph.		
Pre-Requisite			
1.	Mathematics –I (BS(CS/IT)101, Mathematics-III(BS(CS/IT)307)		
2.	Engineering Mathematics (UG level)		
Unit	Content	Hrs	Marks
1	Module 1: Theory of Numbers: Principles of Mathematical Induction, Well Ordering Principle, Divisibility theory and properties of divisibility; Fundamental theorem of Arithmetic; Euclidean Algorithm for finding G.C.D and some basic properties of G.C.D with simple examples; Congruence, Residue classes of integer modulo n (Z_n) and its examples, Chinese Remainder Theorem.	7	
2	Module 2: Counting Techniques: Pigeon- hole Principle, Principles of inclusion and exclusions; Recurrence relations: Formulation & Modelling of different counting problems in terms of recurrence relations, Solution of linear recurrence relations with constant coefficients (upto second order) by (i) The iterative method (ii) Characteristic roots method (iii) Generating functions method.	6	
3	Module 3: Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. Proof Techniques: Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency. Disjunctive and Conjunctive normal form.	7	
4	Module 4: Algebraic Structures and Morphism:	10	

	Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Permutation Groups, Normal Subgroups, Quotient group, Homomorphism & Isomorphism (Elementary properties only). Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function.		
5	Module 5: Graph Theory: Planar and Dual Graphs. Kurowski's graphs. Homoeomorphic graphs. Eulers formula ($n - e + r = 2$) for connected planar graph and its generalisation for disconnected graphs. Detection of planarity. Graph colouring. Chromatic numbers of simple graphs. Chromatic Numbers and its bounds, Independence and Clique Numbers, Perfect Graphs-Definition and examples, Chromatic polynomial and its determination, Applications of Graph Colouring. Simple applications of chromatic numbers. Statement of four and five colour theorems.	6	

Course outcomes

After completion of the course, a student would be able to:

CO 1	determine multiplicative inverses, modulo n and use to solve linear congruences.
CO 2	solve different engineering problems using counting techniques.
CO 3	express a given logic sentence in terms of predicates, quantifiers, and logical connectives and derive the solution for a given a problem using deductive logic and prove the solution based on logical inference.
CO 4	classify the algebraic structure for a given mathematical problem and evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.
CO 5	develop the given problem as graph networks and solve with techniques of graph theory.

Learning Resources:

1.	C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition by, Tata McGraw – Hill.
2.	N. Chandrasekaran and M. Uma Parvathi, Discrete Mathematics, PHI.
3.	J.K. Sharma, Discrete Mathematics, Macmillan.
4.	Malik, Margeson, Sen, Fundamentals of abstract algebra, Tata McGraw-Hill.
5.	Clark John, Holton Derek Allan, A First Look at Graph Theory, World Scientific.

Name of the course	Design and Analysis of Algorithm
Course Code: PC(CS)408	Semester: 4th
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory: 3 hrs./week	Mid Term Exam I: 15 Marks

Tutorial: NIL		Mid Term Exam II: 15 Marks	
Practical: NIL		Assignment.: 20 Marks	
Credit Points: 3		Semester End Exam: 75 Marks (Two third weightage for final reckoning i.e., 50 marks)	
Objective:			
1.	To understand different paradigms of algorithms such as greedy, dynamic programming, divide and conquer etc..		
2.	To calculate the time complexities of algorithms.		
3.	The ability to decide based on a given problem which design paradigm and algorithm is appropriate		
Pre-Requisite:			
1.	Data Structure and Algorithm (PC(CS)302)		
Module	Content	Lecture Hours	
1	Non-comparison-based sorts - Radix sort, count sort; Median order statistics; Lower bound of sorting.	6	
2	Algorithm design frameworks - Divide/Decrease and Conquer, Backtracking, Greedy, Dynamic Programming, Decision and Optimization problems; Comparison - Divide & Conquer, Greedy and Dynamic Programming. Illustrations of various design framework: Dynamic Programming - Optimal substructure and overlapping sub problems; Matrix-chain multiplication; Backtracking - 8-queens problem; Greedy Method - Knapsack problem, Job sequencing with deadlines.	10	
3	Disjoint set data structure.	6	
4	Graph Algorithms: BFS and DFS- algorithm and comparison; Prim's and Kruskal's algorithms for finding minimum spanning tree, Single source shortest path, All pair shortest paths; Network Flow.	10	
5	Deterministic and non-deterministic algorithms, Tractable and Intractable problems, Solvability, P and NP: Notion of NP Class: P, NP, NP-hard, NP-complete; reduction (concept only); Cook's theorem (statement only).	4	
Course Outcomes:			
After completion of this course, the learners will be able to-			
CO1	Implement non-comparison-based sorts		
CO2	Analyze a problem to determine which design paradigm to use to solve the problem.		
CO3	Apply disjoint set data structures		
CO4	Solve various graph problems efficiently.		

CO5	Identify whether a problem is in P or NP
Learning Resources:	
1	T. H. Cormann, C.E. Leiser son, R. L. Rivest and C. Stein, "Introduction to Algorithms", PHI.
2	Ellis Horowitz, Sartaj R. Sahani, "Fundamentals of Computer Algorithms". Computer Science Press.
3	A. Aho, J. Hopcroft and J. Ullman, "The Design and Analysis of algorithms", Pearson Education.
4	D.E. Knuth: The Art of Computer Programming, Vol. 1, Vol. 2 and Vol. 3, Addison-Wesley.
5	G. Brassard, P. Bratley, Fundamentals of Algorithmics -, PHI.
6	S. Baase, Allen Ven Gelder "Computer Algorithms-Introduction to Design & Analysis"- 3 rd Edition, Pearson Education

Name of the course		Formal Language and Automata Theory	
Course Code: PC(CS)409		Semester: 4th	
Duration: 6 month		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 3hrs/week		Mid Term 1 Exam: 15 Marks	
Tutorial: 1hrs/week		Mid Term 2 Exam: 15 Marks	
Practical: 0 hrs/week		Other Assessment tools (Assignment, Quiz etc.): 20 Marks	
Credit Points: 4		End Semester Exam: 100 Marks (50% weightage for final reckoning i.e., 50 mark)	
Objective:			
1.	Introduce formal models of computation and the concept of language recognition.		
2.	Develop a theoretical foundation for understanding the limits of computation.		
3.	Analyze grammars, automata, and computational complexity hierarchies.		
4.	Familiarize students with undecidable problems and the capabilities of Turing Machines.		
Pre-Requisite			
1.	Discrete Mathematics		
2.	Mathematical Reasoning and Problem Solving		
Unit	Content	Hrs	Marks
1	Module 1: Mathematical Foundations and Finite Automata Set theory, relations, functions, formal proofs (contrapositive, contradiction, counterexamples), Alphabets, strings, languages, grammar formalism, Chomsky hierarchy, Deterministic and Non-Deterministic Finite Automata (DFA, NFA, ϵ -NFA), Finite Automata with output (Moore and Mealy), Conversions: NFA \rightarrow DFA, ϵ -NFA \rightarrow DFA, Minimization of FA, Myhill-NE rode Theorem	7	
2	Module 2: Regular Languages and Expressions Regular Expressions (RE) and FA, Algebraic laws for REs, Arden's Theorem, Constructing RE from FA and vice versa, Pumping Lemma for RL, Applications Equivalence of FA and RE, Regular grammars and conversion between RE and regular grammars, Closure properties, Decision problems	7	

3	Module 3: Context-Free Grammars and Languages Context-Free Grammars (CFG): definition, derivation, parse trees, Ambiguity, left recursion, left factoring, Simplification of CFGs: useless symbols, null and unit productions Normal forms: CNF, GNF Closure properties, Pumping Lemma for CFLs Decision problems for CFLs Relationship between CFG and Regular Languages	7	
4	Module 4: Pushdown Automata Introduction and formal definition, Instantaneous descriptions and transitions, Acceptance by empty stack and final state, Equivalence between PDA and CFG, Deterministic PDA and Non-deterministic PDA, Two-stack PDA	6	
5	Module 5: Turing Machines and Undecidability Basics and formal definition of TMs, Transition diagrams and instantaneous descriptions, Variants of TMs: Multi-tape, NDTM, Universal TM, TM as an integer function, LBA, Unrestricted grammars, Recursive and Recursively Enumerable Languages, Undecidability and Reducibility, Post's Correspondence Problem (PCP), Modified PCP	9	

Course outcomes

After completion of the course, a student would be able to:

CO 1	Understand and construct formal machine models and grammars.
CO 2	Analyse the recognition power and equivalence of finite automata and regular expressions.
CO 3	Design and simplify context-free grammars and apply parsing techniques.
CO 4	Model and simulate pushdown automata for context-free languages.
CO 5	Analyse Turing machines and understand the concept of decidability and undecidability.

Learning Resources:

1.	Hopcroft, Motwani, Ullman – Introduction to Automata Theory, Languages and Computation, Pearson
2.	Peter Linz – An Introduction to Formal Languages and Automata, Jones & Bartlett
3.	Martin, J.C. – Introduction to Languages and the Theory of Computation, McGraw Hill

4.	Mishra, Chandrasekaran – Theory of Computer Science: Automata, Languages and Computation, PHI
5.	Michael Sipser – Introduction to the Theory of Computation, Cengage

Name of the course		Computer Architecture	
Course Code: PC(CS) 410		Semester: 4th	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 3 hrs./week		Mid Term I: 15 Marks	
Tutorial: 1 hrs/week		Mid Term II: 15 Marks	
Credit Points: 4		Assignment, Test based on assignments, Surprise tests, Quizzes, Presentations, etc. : 20 Marks	
		End Semester Exam: 50 Marks	
Objectives:			
1.	To identify different processor architectures and their performance measurement parameters.		
2.	To apply different techniques for improving the performance of processor.		
3.	To develop the concept of multiprocessor architecture.		
4.	To design pipeline processor architecture.		
Pre-Requisites:			
1.	Digital Electronics [ES(CS/IT)307]		
2.	Computer Organization [PC(CS/IT)301]		
Unit	Content	Hrs	Marks
1	Pipelining Architecture: Introduction: Review of basic computer architecture (Revisited), Quantitative techniques in computer design, measuring and reporting performance. Pipelining: Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards and structural hazards, techniques for handling hazards. Exception handling. Pipeline optimization techniques.	9	30
2	Instruction-level parallelism: Basic concepts, techniques for increasing ILP, RISC Architecture, superscalar, super pipelined and VLIW processor architectures. Array and vector processors.	9	20
3	Multiprocessor architecture: taxonomy of parallel architectures; Centralized shared-memory architecture: synchronization, memory consistency, interconnection networks. Distributed shared-memory architecture, Cluster computers. Non von Neumann architectures: data flow computers, reduction computer architectures, systolic architectures	9	30

4	Memory Module: Hierarchical memory technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses, cache mapping techniques; Virtual memory organization.	9	20
Course Outcomes: After completion of this course students will be able to-			
CO1	Explain the concept of pipeline architecture, different hazards and analyze different techniques for handling pipeline hazards		
CO2	Describe and differentiate non-von Neumann architectures such as data flow, reduction, and systolic architectures and their applications in parallel computation.		
CO3	Classify and compare different multiprocessor architectures, interconnection networks, and synchronization mechanisms in shared and distributed memory systems.		
CO4	Evaluate instruction-level parallelism and compare architectural approaches such as RISC, superscalar, super pipelined, and VLIW for exploiting ILP.		
CO5	Analyse memory hierarchy designs including cache and virtual memory systems, and apply techniques to reduce memory access latency and cache misses.		
Learning Resources:			
1.	Advanced Computer Architecture-Kai Hwang & Naresh Jot Wani, McGraw Hill		
2.	Computer Architecture and Parallel Processing -Kai Hwang and A. Briggs, McGraw Hill		
3.	Computer Architecture: a quantitative approach - J. L. Hennessy and D. A. Patterson, Harcourt Asia, Singapore.		
4.	Computer Organization and Architecture - V. Rajaraman and T. Radhakrishnan PHI Learning Pvt Ltd.		
5.	Computer Architecture and Parallel Processing - Hwang and Briggs, TMH.		
6.	Computer Architecture and Organization - Hayes, McGraw-Hill.		

Name of the course		Operating systems	
Course Code: PC(CS)411		Semester: 4th	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: 3 hrs./week		Mid Term I Exam: 15 marks	
Tutorial: 1 hr./week		Mid Term II Exam: 15 marks	
Credit Points: 4		Assignment, Test based on assignments, Surprise tests, Quizzes, Presentations, etc.: 20 Marks	
		End Semester Exam: 100 Marks	
Objective:			
1.	To understand and analyze operating system structures and services.		
2.	To understand and determine Process management in Operating System.		
3.	To understand and determine Memory management and file management in Operating System.		
4.	To analyze and assess Disk management, I/O management and protection & security in Operating System.		
Pre-Requisite			
1	Computer Organization – PC(CS)301		
2	Data Structures & Algorithms - PC(CS)302		
Unit	Content	Hrs	Marks
1	Introduction of O.S: Concept of OS. Operating system services, dual-mode operation, Evaluation of O.S, Different types of O.S: batch, multi-programmed, timesharing, real-time, distributed, network. Introduction of Process: Concept of process, Process life cycle, Resource utilization, Operations on processes, IPC.	4	
2	System Structure: Computer system operation, Operating system structure, kernel: microkernel, monolithic kernel, system calls. Threads: Overview, Benefits of threads, User and kernel threads, multithreading models.	4	
3	CPU Scheduling: Scheduling criteria, Preemptive & non-preemptive scheduling, Scheduling algorithms (FCFS, SJF/SRTF, RR, Priority), MLQ scheduling, multi-processor scheduling. Process Synchronization: Race condition, Critical Section problem, Semaphore, Mutex, Monitor. Deadlocks: Deadlock criteria, Methods for handling deadlocks, Resource allocation graph, Banker's algorithm, Recovery from deadlock.	10	
4	Memory Management: Background, Logical vs. physical address, Address binding, Swapping, Contiguous memory allocation, Fragmentation, Segmentation, Paging. Virtual Memory: Concept, Demand paging, Page replacement, Page replacement algorithms (FCFS, LRU, Optimal). File Systems: File attributes, File system structure, File access methods, File allocation methods (contiguous, linked, indexed).	8	
5	Disk Management: Disk structure, Disk formatting, Boot block, Bad blocks, Disk scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK).	3	

6	I/O Management: I/O hardware, Polling, Interrupts, DMA, Application I/O interface, Kernel I/O subsystem, Spooling and device reservation. Protection & Security: Goals of protection, Security problem, Authentication, Program threats, System threats	7	
Course Outcome: After completion of the course students will able to			
CO1	analyse different types of operating system.		
CO2	Select different types of kernels in operating system		
CO3	Apply different mechanism to handle process management		
CO4	Determine different memory management, file management mechanism to provide better performance to users		
CO5	Evaluate different disk management policies		
CO6	Implement different techniques for protection and security.		
Learning Resources			
1	Operating System concepts- A. Silberschatz, Greg Gagne, and Peter Baer Galvin- Wiley India		
2	Operating Systems: Internals and Design Principles-William Stallings-Pearson		
3	Operating Systems Concepts & design - Milan Milenkovic, TMH		
4	Tanenbaum A.S. and Woodhull “Operating System Design & Implementation”, Pearson		
5	Advanced Concepts in operating Systems - Mukesh Singhal and Niranjana G. Shivaratri, TMH		
6	Operating System. - Dhamdhare: - TMH		
7	An Introduction to Operating Systems- Dietel H. N- Addison Wesley.		

Name of the course:		Algorithm Lab	
Course Code: PCL(CS)412		Semester: 4 th	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: NIL		Attendance 10	
Tutorial: NIL		Preparation of Lab Report: 30	
Practical:3 hrs/week		Experimental data/Precision of work done: 30	
Credit Point:1.5		Presentation/ analysis of the result: 10	
		Viva Voce:20	
Objective:			
1.	To understand the working of Fundamental algorithms such as sorting.		
2.	To analyse the performance of algorithms based on the underlying data structures		
3.	To implement various graph algorithms		
4.	To decide which algorithms to employ based on nature of problem.		
Pre-Requisite:			
1.	NA		
Unit	Content	Hours	Marks
1	Comparison of performance of various sorting algorithms.	03	
2	Implementation of median order statistics in O(n) time	03	
3	Performance comparison of problem solving using dynamic programming and recursion.	06	
4	Implement disjoint set data structure and compare the performance based on different data structures used.	03	
6	Implementation of BFS and DFS both recursive and non-recursive version and their performance comparison.	03	
7	Implementation of Prim's algorithm and performance comparison based on different data structures used	03	
8	Implementation of Dijkstra's algorithm and performance comparison based on different data structures used	03	
9	Implementation of Bellman Ford algorithm and all pair shortest path algorithm	03	
10	Implementation of Network flow	03	
Course Outcomes:			
After completion of this course, the learners will be able to-			
CO1	Compare performance of various sorting algorithm.		
CO2	Decide which design paradigm to use for a particular problem		
CO3	Implement various graph algorithms		
CO4	Apply graph algorithms to real life problems		

CO5	Compare performance of recursion and dynamic programming
Learning Resources:	
1.	T.H. Cormen, C.E. Leiserson, R. Rivest and C. Stein: <i>Introduction to Algorithms</i> , (Second/Third Edition), PHI, 2009.
2.	R. Sedgewick: <i>Algorithms in C</i> , Pearson, 2004.
3.	Steven S Skiena, <i>Algorithm design manual</i> , 2 nd Edition, Springer.

Name of the Course		Programming Lab using C++	
Course Code: PCL(CS)413		Semester: 4th	
Duration: 6 months		Total Marks: 100	
Teaching Scheme		Examination Scheme	
Theory: Nil		Attendance: 10 marks	
Tutorial: Nil		Preparation of Lab Report: 30 marks	
Practical: 3 hrs/week		Experimental data/ Precision of work done: 30 marks	
Credit Points:1.5		Presentation / analysis of the result: 30 marks	
		Viva voce: 20 marks	
Objective			
1. To learn the object-oriented programming paradigm			
2. To incorporate data security or protections issues in object-oriented programming through C++			
3. To implement data abstraction scenario through C++ programming			
4. To implement dynamic resolving of method call scenarios at run time situations			
5. To handle run time exceptional scenarios through C++ programming			
Prerequisite			
C programming			
Detailed Syllabus			
Module	Content	Hrs	Marks
1	Basic idea of object-oriented programming, basic C++ programme with class, object, I/O streams, constructor, destructor, function overloading, copy constructor, identification of variables & methods with scope resolution operator, alias variable concepts,	6	
2	Dynamic memory allocation for variables and objects, array of objects, pointer to class data members, functions and objects	3	
3	Dealing with static variables & methods, static and non-static inner classes, local classes	6	
4	Friend functions- partial & full friend declarations, operator overloading – binary and unary operator overloading, operator overloading with friend and class member function	6	
5	Inheritance, base & derived class, hierarchical, multiple, multilevel inheritance with constructor calling sequence, derived class constructor declaration	3	
6.	Virtual base class concept, implementation of function overriding, static and dynamic resolving of function calls, pointers to base & derived classes, virtual functions and run time polymorphism, abstract class implantation	3	
7.	Basic concepts of exception handling with try/catch, class and function templates	3	

Course outcomes

After completion of the course, a student would be able to:

CO 1	Demonstrate C++ program with class, object and constructor concepts
CO 2	Apply different data & class member protection approaches through C++ programming
CO 3	Apply inheritance and various abstraction concepts through C++ programming
CO 4	Implement various types of polymorphisms through C++ programming
CO 5	Design effective C++ program codes by taking care of exceptional scenarios and template concepts

Name of the course		Operating System Lab	
Course Code: PCL(CS)414		Semester: 4th	
Duration: 6 months		Maximum marks:100	
Teaching Scheme		Examination scheme:	
Theory: Nil		Attendance: 10 marks	
Tutorial: Nil		Preparation of Lab Report: 30 marks	
Practical: 3 hrs./week		Experimental data/ Precision of work done: 30 marks	
Credit Points:1.5		Presentation / analysis of the result: 30 marks	
		Viva voce: 20 marks	
Unit	Content	Hours	Marks
1.	Familiarization of Linux Commands.	3	
2.	Shell in UNIX. Different types of Shell in UNIX. Creating a bash shell script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).	6	
3.	Implementation of CPU scheduling algorithms.	3	
4.	Implementation of classical problems in process synchronization.	6	
5.	Implementation of deadlock handling techniques.	6	
6.	Implementation of memory management techniques.	3	
7.	Operations on Processes, signals, Pipes and system calls.	3	
Course Outcomes:			
After completion of the course students will able to -			
CO1	Review commands in UNIX.		
CO2	Write programs using shell scripts.		
CO3	Implement different process management mechanisms.		
CO4	Implement different memory management techniques.		
CO5	Evaluate different system management mechanisms.		

Name of the course		Computer Architecture Lab	
Course Code: PCL(CS)415		Semester: 4th	
Duration: 6 months		Maximum marks:100	
Teaching Scheme		Examination scheme:	
Theory: Nil		Attendance: 10 marks	
Tutorial: Nil		Preparation of Lab Report: 30 marks	
Practical: 3 hrs./week		Experimental data/ Precision of work done: 30 marks	
Credit Points:1.5		Presentation / analysis of the result: 30 marks	
		Viva voce: 20 marks	
Unit	Content	Hours	Marks
1.	Introduction to Logisim / HDL	03	
2.	Basic Logic Gates and Combinational Circuits	06	
3.	Arithmetic Circuits	03	
4.	Counter design	03	
5.	Register design	03	
6.	Memory unit design and perform memory operations.	03	
7.	Simple ALU design	03	
8.	Simple CPU design	03	
9.	Interfacing of CPU and Memory	03	
Course Outcomes:			
After completion of this course the students will be able to -			
CO1	Construct truth tables and simulate circuits using logic gates, adders, multiplexers, and encoders in Logisim.		
CO2	Design registers and counters, and understand their role in data storage and control logic.		
CO3	Build functional ALUs capable of performing basic arithmetic and logical operations.		
CO4	Implement simple memory units and understand data flow within computational systems.		
CO5	Integrate ALU, memory, control logic, and registers to build and simulate a basic CPU using Logisim.		
Learning Resources:			
1	Octave online https://octave-online.net/ the open-source alternative for simulation of the above experiments		

5th SEMESTER

SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs./WEEK	CREDIT
THEORY							
01	PC(CS)516	Software Engineering	3	0	0	3	3
02	PC(CS)517	Computer Network	3	1	0	4	4
03	PC(CS)518	Object Oriented Programming	3	1	0	4	4
04	PC(CS)519	Artificial Intelligence	3	0	0	3	3
05	PEC(CS)501	Elective-I A. Soft Computing B. Signal & Systems	3	0	0	3	3
06	MC(CS)502	Constitution of India	2	0	0	2	2
SESSIONAL/PRACTICAL							
01	PCL(CS)520	Computer Networking Lab	0	0	3	3	1.5
02	PCL(CS)521	Programming Lab using Java	0	0	3	3	1.5
03	PROJ(CS)502	Minor Project 2 (Model Design)	0	0	2	2	1
04	TRN-SSD(CS)502	Summer Training 2 (Inter- institutional / Industrial)	0	0	0	0	1
TOTAL			17	2	08	27	24

Name of the course	Software Engineering		
Course Code: PC(CS)516	Semester:5 th		
Duration:6months	MaximumMarks:100		
Teaching Scheme	Examination Scheme		
Theory:3hrs/week	MidSemester1 Exam:15Marks		
Tutorial:0hrs/week	MidSemester2Exam:15Marks		
Practical:0hrs/week	Other Assessment tools (Assignment, Quiz etc.):20Marks		
Credit Points: 3	End Semester Exam:(100marks converted to 50):50Marks		
Objective:			
1.	To learn the different models for the development of a software product		
2.	To explore the designing, coding and testing to develop software product		
3.	To asses quality of software product to sustain in the market		
Pre-Requisite			
1.	Fundamentals of Programming knowledge, C, C++ etc.		
Module	Content	Hours	Marks.
1	Software Development Process Models: Waterfall,Spiral,Prototyping,RAD,Evolutionary, Software Requirement and Feasibility Analysis, Cost- Benefit Analysis,	05	
2	Software Design: Context Diagram, DFD, Data Dictionary, ER diagram, Decision Tree, Decision Table, Structured Chart, Structured English, Top-Down and Bottom-Up design, Modular Programming, Module Relationship-Coupling, Cohesion,	06	
3	Object Oriented Software Designing with UML: Functional vs. Object- Oriented approach. Basic idea of Object-Oriented Modelling & its purpose, Primary goals of UML, class, object and interface diagram, brief idea of use case diagram, sequence diagram, collaboration diagram, state chart and transition diagram, activity and deployment diagram,	06	
4	Software Quality: Quality Attributes, Total Quality Management, Software Quality Assurance and Quality Control, Software Quality metrics, Reliability, MTTF, MTBFReliability Models. Software Reuse, Software Maintenance	06	
5	Software Testing: Software Testing Strategy, Unit Testing, Integration Testing, System Testing, Debugging, White-Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing, Verification, Validation, Cyclomatic Complexity.	06	
6	Software Project Management: Software Project Management Concepts, Software Project Management Plan, Tools for Project Plan – WBS, PERT, GANTT, Project Scheduling & Monitoring, Staffing, Cost Estimation, COCOMO, Software Reengineering Process model, Software Configuration Management (SCM), Case study on Software Development Process etc.	07	

Course outcomes	
After completion of the course, a student would be able to:	
CO 1	Understand different types of models.
CO 2	Analyse different design approaches
CO 3	Create object-oriented software design with UML
CO 4	Evaluate the quality of software
CO 5	Apply various testing techniques
CO 6	Apply the activity of software project management with CASE study.
Learning Resources:	
1.	Software Engineering: A practitioner's approach– Pressman (TMH)
2.	Software Engineering: Pankaj Jalote (Wiley-India)
3.	Software Engineering: Rajib Mall (PHI)
4.	Software Engineering: Agarwal and Agarwal, (PHI)
5.	Software Engineering: Sommerville, Pearson
6.	Fundamentals of Software Engineering – C. Ghezzi, M. Jazayeri, D. Mandrioli
7.	Software Engineering Martin L. Shooman,– TMH

Name of the course:		Computer Network	
Course Code: PC(CS) 517		Semester: 5 th	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme	
Theory Contact Hrs.: 3 hrs/week		Mid Semester-1 Exam: 15 Marks	
Tutorial Contact Hrs.: 1 hrs./week		Mid Semester-2 Exam: 15 Marks	
Credit Point: 4		Assignment, Quiz & class attendance: 20 Marks	
		End Semester Exam: 75 Marks (to be mapped into 50 marks)	
Objective:			
1.	To study the concept of computer network and protocol suite		
2.	To study Physical and data link layer and related hardware and protocol		
3.	To study network layer, routing protocols, IP addressing		
4.	To study transport layer, TCP and socket		
5.	To study Application layer and network security		
Pre-Requisite			
1.	Communication engineering (PC(CS)303)		
Module	Content	Hrs	Marks
1.	Introduction: Overview of Data Communication and Networking; Layered Network Architecture; Mode of communication, topology, Data and Signal; categories of network (LAN, MAN, WAN); Internet: brief history, Protocols and standards; Reference models: OSI reference model, TCP/IP reference model, their comparative study.	03	
2.	Physical Layer: Transmission Media: Guided, Unguided; switching: time division & space division switch, TDM bus, Banyan switch; MODEM, Repeater and hub, The telephone network system.	02	
3.	Data link Layer: Error Control: Types of errors, framing (character and bit stuffing), error detection & correction; Flow control: Protocols: Stop & wait ARQ, Go-Back- N ARQ, Selective repeat ARQ, HDLC; Medium Access sub layer: Point to Point Protocol, LCP, NCP, Token Ring; Access mechanism: Reservation, Polling, Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA, Traditional Ethernet, fast Ethernet.	07	
4.	Network layer: Internetworking & devices: Bridges, Switches, Router, Gateway; Addressing: IP addressing (IPV4, IPv6), masking, Classful and Classless Addressing, Subnetting, NAT; Routing : Intra and Inter Domain Routing, Unicast, Multicast, Broadcast routing. static vs. dynamic routing, Unicast Routing Protocols: RIP, OSPF, BGP; Other Protocols: ARP and RARP, IP, ICMP, IPV6; Mapping between IP and MAC address: ARP & RARP Switching, Communication Networks: Circuit switching; Packet switching; Routing in packet switched networks; X.25; Frame Relay; ATM, SONET.	10	
5.	Transport layer : Process to Process delivery; UDP; TCP, Features, Segment, Three-Way Handshaking, socket and port addressing, Flow Control, Error Control, Congestion Control: Open Loop, Closed Loop, choke packets; Quality of service: techniques to improve QoS: Leaky bucket algorithm, Token bucket algorithm.	08	
6.	Application Layer : Introduction to DNS, SMTP, SNMP, FTP, HTTP & WWW; Security: Attacks, Cryptography, Firewalls, IDS & IPS, Malware, IP and transport layer security, DMZ. Data Communication Technologies: ISDN services & ATM, DSL technology, Wireless LAN, Bluetooth, VPN.	10	

Course Outcome:	
After completion of this course the students will be able to -	
CO1	investigate two protocol suits and different topologies, transmission media of computer network
CO2	investigate different random and controlled access mechanism , flow and error control
CO3	assess different routing models for computer network and IP addressing
CO4	assess quality of services (Qos) in Transport layer
CO5	investigate different security protocols and encryption mechanism
CO6	design of real life problems and solution for IP addressing and DMZ
Learning Resources:	
1.	B. A. Forouzan – “Data Communications and Networking (3rd Ed.)” – TMH
2.	A. S. Tanenbaum – “Computer Networks (4th Ed.)” – Pearson Education/PHI
3.	W. Stallings – “Data and Computer Communications (5th Ed.)” – PHI/ Pearson Education
4.	Black, Data & Computer Communication, PHI
5.	Kurose and Rose – “Computer networking -A top down approach featuring the internet” – Pearson Education

Name of the course:	Object Oriented Programming through Java
Course Code: PC(CS) 518	Semester: 5 th
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory Contact Hrs.: 3 hrs/week	Mid Semester-1 Exam: 15 Marks
Tutorial Contact Hrs.: 1 hrs./week	Mid Semester-2 Exam: 15 Marks
Credit Point: 4	Assignment, Quiz & class attendance: 20 Marks
	End Semester Exam: :(100marks converted to 50):50Marks

Objective:

1. To understand critical aspects of Object-Oriented Programming (OOP) in the context of basic Java Programming
2. To implement code reusability and abstraction scenarios of OOP by way of Java Programming
3. To construct modular and GUI based java programming for interactive user applications with data storage
4. To handle different exceptional cases in Java Programming.
5. To implement parallel program execution scenarios through Java Programming

Pre-Requisite

- 1.

Mod ule	Content	Hrs	Marks
1.	Module-I: OOP Paradigm & Java Programming Basics Features of OOP, Object Oriented Modelling - class diagram, linking, association, aggregation, generalization & specialization. comparison of OOP & procedural programming, design & execution of java program, JDK & JRE, constructor, method overloading, call by value & call by reference, garbage collection, use of this keyword, static variables & methods, nested class & its types, string handling & manipulation operations, command line arguments	08	
2.	Module-II: Code Reusability, Abstraction & Packages in Java Inheritance, super & sub class member accessing rules, use of super keyword, superclass constructor call from subclass, method overriding, dynamic polymorphism, abstract class, interface implementation, combining abstract classes & interfaces, extending interface, multi-interface implementation, modularity with packages, access protection in packages, importing of packages.	08	
3.	Module-3: Exception Handling in Java Need for exception handling, exception class hierarchy, concept of try & catch block, nested try statements, use of multiple catch clauses, error & exception, throw & throws clause in java and their differences, compiler based and run time exception, use of finally block, construction of user defined exception class	08	
4.	Module 4: Multithread Programming with Java advantage of multithreading, concept of main thread & child thread, thread life cycle, creation of multiple child threads- implementing runnable interface & extending Thread class, setName(), getName(), sleep(), isAlive() & join() method, thread synchronization, interthread communication	08	
5.	Module 5: GUI, Data Storage concepts in java & JDBC Advanced layout management: Model-View-Controller properties, concept of swing, JOptionPane package with input and output utilities & dialog boxes, basic file handling concepts in java with data storage and reading, JDBC concepts, Database drivers, relational SQL execution via Connection Statement interfaces,	04	

	parameter bindings, and ResultSet rows		
Course Outcome:			
After completion of this course the students will be able to -			
CO1	Explain basic properties of OOP and structure of a Java Program		
CO2	Apply code reusability and abstraction features of java in modular programming		
CO3	Discuss different exception handling mechanisms that may be required for execution of java program		
CO4	Describe multithread programming in java for performing parallel execution of multiple tasks		
CO5	State data storage and retrieval concept with user interactive java program		
Learning Resources:			
1.	Core Java Volume I — Fundamentals (9th Edition) by Cay S Horstmann and Gary Cornell		
2.	Rambaugh, James Michael, Blaha, Object Oriented Modelling and Design, Prentice Hall, India		
3.	Java: A Beginner’s Guide by Herbert Schildt, Oracle Press.		
4.	Deitel and Deitel- “Java How to Program”, Pearson Education.		
5.	Object Oriented Programming with Java, R. Buyya, S.T. Selvi, X. Chu, Tata McGraw Hill (TMH).		
6.	Introduction to Java Programming (Comprehensive Version), Y. Daniel Liang, Pearson Education.		
7.	Core Java Volume I — Fundamentals (9th Edition) by Cay S Horstmann and Gary Cornell		
8.	Rambaugh, James Michael, Blaha, Object Oriented Modelling and Design, Prentice Hall, India		
9.	Java: A Beginner’s Guide by Herbert Schildt, Oracle Press.		

Name of the course	Artificial Intelligence
Course Code: PC(CS)519	Semester: 5th
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory Contact Hrs.: 3 hrs/week	Mid Semester-1 Exam: 15 Marks
Tutorial Contact Hrs.: 0 hrs./week	Mid Semester-2 Exam: 15 Marks
Practical Contact Hrs.: 0 hrs./week	Assignment, Quiz & class attendance: 20 Marks
Credit Points: 3	End Semester Exam: 100 Marks (to be mapped into 50 marks)
Objective:	
1	Understand the critical role of AI with fundamental concept of knowledge representation & Inferencing in AI
2	Formulate a problem as state space exploration framework with principles, models & algorithms of AI
3	Use AI-heuristic strategies to find acceptable solutions for searching, reasoning & language processing.
4	Realizing problems under uncertainty and plan mechanisms to deal with them through allied branches of AI
Pre-Requisite	
1	Mathematics
2	Data structure & Algorithm

Module	Content	Hrs.	Marks
1	MODULE-1: Introduction of AI and Agents: Overview and foundation of AI, goals of AI, Problems of AI, AI environment & its nature, AI agent & structure, different types of AI agents- goal based agents, utility-based agents, learning agents, relation between AI environment and agent, concept of PEAS	04	
2	MODULE-2: Problem Solving & Searching: basic rules in AI for problem-solving, problem-solving agents, problem solving in AI based on search techniques, searching for solutions, uniform search strategies: generate and test, breadth first search, depth first search, bidirectional search, comparing uniform search strategies, heuristic search, greedy best-first search, A* search, Hill-climbing search.	08	
3	MODULE-3: Knowledge & Reasoning: knowledge-based agent, concept of logic, Propositional Logic, First Order Logic: Syntax and Semantics, Inference in First Order Logic, propositional vs first order inference, knowledge engineering in first order logic: Unification and Lifting, forward and backward chaining.	08	
4	MODULE-4: Planning and Learning: classical planning, uncertainty planning, hierarchical planning, planning & acting in nondeterministic domain, Multi agent Planning, bayes' rule & its use, different forms of learning, human & machine learning, supervised, unsupervised & reinforcement learning, neural network & genetic learning.	08	
5	MODULE 5: Applications & Ethics of AI: Different applications of AI, concept of natural language processing, language models, text classification, information retrieval, concept of expert Systems: characteristics & capabilities, human-computer interaction, idea of perception, computer vision, speech recognition, ethics in AI, future of AI	08	

Course Outcome:

After completion of this course the students will be able to -

CO1	Explain the importance as well as main functionalities and techniques of AI
CO2	Discuss different searching techniques in the context of problem solving through AI
CO3	Demonstrate the concept of knowledge representation and reasoning techniques in AI
CO4	Describe different types of planning and Learning Systems in AI
CO5	Recognize the applications & ethics of AI in the context of technological advancements

Learning Resources:

1	Introduction to Artificial Intelligence, E. Charniak, et.al., PEARSON
2	Artificial Intelligence, P. H. Winston, PEARSON
3	Artificial Intelligence, E. Rich and K. Knight, TMH.
4	“Artificial Intelligence: A Modern Approach” Stuart Russell, Peter Norvig, 2nd Edition, Prentice Hall

Name of the course:	Soft Computing		
Course Code: PEC(CS) 501A	Semester: 5 th		
Duration: 6 months	Maximum Marks: 100		
Teaching Scheme	Examination Scheme		
Theory Contact Hrs.: 3 hrs/week	Mid Semester-1 Exam: 15 Marks		
Tutorial Contact Hrs.: 0 hrs./week	Mid Semester-2 Exam: 15 Marks		
Credit Point: 3	Assignment, Quiz & class attendance: 20 Marks		
	End Semester Exam:100 Marks (to be mapped into 50 marks)		
Objective:			
1.	To understand basic soft computing techniques		
2.	To learn how to use soft computing technique for a particular problem		
3.	To implement hybrid soft computing techniques		
Pre-Requisite:			
1.	Discrete Mathematics BS(CS/IT)408		
2.	Design and Analysis of Algorithm PC(CS/IT)406		
Module	Content	Hrs.	Marks
1	Introduction to Soft Computing: Concept of computing systems, "Soft" computing versus "Hard" computing, Characteristics of Soft computing, some applications of Soft computing techniques.	02	
2	Artificial Neural Network: Introduction to Artificial Neural Networks, Perceptron, Neural Networks Learning Rules, Activation Functions, and Derivation of generalized delta learning rule (back propagation) for Multilayer perceptron. Kohonen Self- Organizing Feature Maps.	12	

3	Fuzzy Logic: Fuzzy Sets, Basic Definitions and Terminology, membership function Set-theoretic operation. Fuzzy union, intersection and complement, various T-norm and T-conorm operators, Fuzzy Relations. Fuzzy Logic, Approximate Reasoning, Compositional Rule of Inference.	05	
4	Evolutionary Algorithms: Genetic Algorithms: Simple GA, Encoding Techniques, Crossover, mutation, inversion and deletion, genetic algorithms in search and optimization. Ant Colony Optimization(ACO). Particle Swarm Optimization(PSO).	10	
5	Hybrid Systems: Hybrid Systems, GA based Fuzzy Systems and Neural Networks Training, Any other applications of soft computing.	07	
Course Outcomes:			
After completion of the course students will able to -			
CO1	Understand the concept of soft and hard computing		
CO2	Compare the relation between real brains and simple artificial neural network models		
CO3	Explain fuzzy sets and represent these sets by membership functions		
CO4	Analyze Evolutionary Algorithms for single and multiple objective optimization problem		
CO5	Design GA based Fuzzy Systems and other hybrid approaches of soft computing techniques for problem solving		
Learning Resources			
1.	S. Rajasekaran and G.A.V.Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI		
2.	"Neuro-Fuzzy and Soft computing", Jang, Sun, Mizutani, PHI		
3.	Neural Networks: A Classroom Approach, 1/e by Kumar Satish, TMH		
4.	Genetic Algorithms in search, Optimization & Machine Learning by David E. Goldberg, Pearson/PHI		
5.	"Fuzzy logic with engineering applications", Timothy J. Ross, John Wiley and Sons.		
6.	"Fuzzy Sets and Fuzzy Logic: Theory and Applications", George J. Klir and Bo Yuan, Prentice Hall		
7.	"Neural Networks: A Comprehensive Foundation (2nd Edition)", Simon Haykin, Prentice Hall.		
8.	"A beginners approach to Soft Computing", Samir Roy & Udit Chakraborty, Pearson		

Name of the course	Signal and Systems		
Course Code: PEC(CS)501B	Semester:5 th		
Duration:6months	MaximumMarks:100		
Teaching Scheme	Examination Scheme		
Theory:3hrs/week	Mid Semester-1 Exam: 15 Marks		
Tutorial:0hrs/week	Mid Semester-2 Exam: 15 Marks		
Practical:0hrs/week	Assignment, Quiz & class attendance: 20 Marks		
Credit Points: 3	End Semester Exam:100 Marks (to be mapped into 50 marks)		
Objective:			
1.	To learn the different types and characteristics of Signal		
2.	To explore linear time variant FS, FT, CT		
3.	To apply random variables and random process		
Pre-Requisite			
1.	Knowledge of Analog and Digital systems, Communication Engineering etc.		
Module	Content	Hours	Marks.
1	Introduction and Signal Characteristics: Signal Types: Continuous-time and discrete-time signals. Signal Classification: Energy and power signals, periodic and aperiodic, even and odd symmetries, deterministic vs. random, and orthogonal signals. Standard Signals: Unit step, unit impulse (Dirac delta), ramp, exponential, and sinusoidal signals. Signal Transformations: Time shifting, time scaling, and folding. System Attributes: Linearity, time-invariance, causality, and stability. System Properties: Memory, causality, stability, time-invariance, linearity, and invertibility.	06	
2	Linear Time-Invariant (LTI) Systems: Impulse Response & Convolution: Convolution integral, graphical convolution, and cascading or interconnecting LTI systems. System Characteristics: Zero-input and zero-state responses, natural and forced responses, and step responses.	05	
3	Frequency Analysis of Continuous-Time Signals: Fourier Series (FS): Representation of periodic signals using complex exponentials, convergence, and the properties of FS.Fourier Transform (FT): Extension of FS to aperiodic signals, properties of FT, and frequency response of LTI systems.	06	
4	Continuous-Time (CT) Systems: LTI Systems: Impulse response and linear convolution. System Equations: Linear constant-coefficient differential equations. Fourier Series & Transform: Representation of periodic and non-periodic signals, Parseval's theorem, and frequency response. Laplace Transform: Definition, Region of Convergence (ROC) and its properties, inverse Laplace transform, and evaluating system stability and causality	06	

	State-Space Analysis: State variable representations of continuous-time systems,		
5.	Random Variables and Random Process: Sets and Sample Spaces Random Variables Continuous and Discrete, Cumulative distribution Function (CDF), Probability Density Function (PDF), Expectation and Moments, Types of Random Processes, Ergodicity, Auto-correlation Function (ACF) & Cross correlation Function (CCF), Power Spectral Density, Wiener– Khinchin–Einstein theorem, Central limit theorem, Transmission of a random process through a Linear Filter.	06	

6.	Discrete-Time (DT) Systems: LTI Systems: Discrete convolution and difference equations. Fourier Analysis: Discrete-Time Fourier Series (DTFS) and Discrete-Time Fourier Transform (DTFT). Sampling Theorem: Concept of sampling and reconstruction .Z-Transform: Definition, ROC properties, poles and zeros, and inverse \mathcal{Z} -transforms. and system analysis. Discrete Transformations: Signal manipulation, discrete-time convolution, and differencing. Sampling Theory: Impulse train sampling, the Nyquist criterion, and the relationship between continuous-time spectra and discrete-time sequences (DTFT).	07	
Course outcomes			
After completion of the course, a student would be able to:			
CO 1	Understand different types of Signals.		
CO 2	Analyse Linear Time-Invariant (LTI) Systems.		
CO 3	Analyse of Continuous-Time Signals.		
CO 4	Evaluate Continuous-Time (CT) Systems.		
CO 5	Apply Random Variables and Random Process.		
CO 6	Apply Discrete-Time (DT) Systems.		
Learning Resources:			
1.	Signals and Systems by A. V. Oppenheim, A. Willsky and S H Nawab,. Prentice Hall,		
2.	Signals and Systems by S. Haskin and B. Van Veen,John Wiley		
3.	Signals & Systems by Simon &Hawkins, Wiley		
4.	Linear Systems and Signals by B. P. Lathi and Roger Green		
5.	Signal Processing and Linear Systemsby B.P.Lathi, Oxford University Press		
6.	Digital Signal Processing: Principles, Algorithms and Applications by Proakis, PHI		
7.	Probability and Random Processes by Venkataraman Krishnan, Wiley & Sons,		
8.	Probability and Random Processes by S.Palani Ammal, PHI Learning,		
9.	Probabilities, Random Variables and Random Signal Principles by P-Z Peebles TMH.		

Name of the course:	Constitution of Indian
Course Code: MC(CS)502	Semester: 5 th
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory Contact Hrs.: 1 hrs/week	Mid Semester-1 Exam: 15 Marks
Tutorial Contact Hrs.: 1 hrs./week	Mid Semester-2 Exam: 15 Marks
Credit Point:	Assignment, Quiz & class attendance: 20 Marks
	End Semester Exam: 100 Marks (to be mapped into 50 marks)

Objective:

1.	To understand the structure of the Indian Constitution.
2.	To make students aware of the theoretical and functional aspects of the Indian Parliamentary System.
3.	To channelize students' thinking towards basic understanding of the legal concepts and its implications for engineers.
4.	To acquaint students with latest intellectual property rights and innovation environment with related regulatory framework
5.	To make students learn about role of engineering in business organizations and e-governance.

Pre-Requisite

1.	Constitution of India MC(CS/IT)502 ,PC (CS/IT)-513
----	--

Module	Content	Hrs	Marks
1	Introduction and Basic Information about Indian Constitution Meaning of the constitution law and constitutionalism, Historical Background of the Constituent Assembly, Government of India Act of 1935 and Indian Independence Act of 1947, Enforcement of the Constitution, Indian Constitution and its Salient Features, The Preamble of the Constitution, Fundamental Rights, Fundamental Duties, Directive Principles of State Policy, Parliamentary System, Federal System, Centre-State Relations, Amendment of the Constitutional Powers and Procedure, The historical perspectives of the constitutional amendments in India, Emergency Provisions: National Emergency, President Rule, Financial Emergency, and Local Self Government – Constitutional Scheme in India.	08	
2	Union Executive and State Executive Powers of Indian Parliament Functions of Rajya Sabha, Functions of Lok Sabha, Powers and Functions of the President, Comparison of powers of Indian President with the United States, Powers and Functions of the Prime Minister, Judiciary – The Independence of the Supreme Court, Appointment of Judges, Judicial Review, Public Interest Litigation, Judicial Activism, LokPal, Lok Ayukta, The Lokpal and Lok ayuktas Act 2013, State Executives – Powers and Functions of the Governor, Powers and Functions of the Chief Minister, Functions of State Cabinet, Functions of State Legislature, Functions of High Court and Subordinate Courts.	08	
3	Introduction and Basic Information about Legal System The Legal System: Sources of Law and the Court Structure: Enacted law -Acts of Parliament are of primary legislation, Common Law or Case law, Principles taken from decisions of judges constitute binding legal rules. The Court System in India and Foreign Courtiers (District Court, District Consumer Forum, Tribunals, High Courts, Supreme Court). Arbitration: As an alternative to resolving disputes in the normal courts, parties who are in dispute can agree that this will instead be referred to arbitration. Contract law, Tort, Law at workplace.	08	

4	Intellectual Property Laws and Regulation to Information Intellectual Property Laws: Introduction, Legal Aspects of Patents, Filing of Patent Applications, Rights from Patents, Infringement of Patents, Copyright and its Ownership, Infringement of Copyright, Civil Remedies for Infringement, Regulation to Information-Introduction, Right to Information Act, 2005, Information Technology Act, 2000, Electronic Governance, Secure Electronic Records and Digital Signatures, Digital Signature Certificates, Cyber Regulations Appellate Tribunal, Offences, Limitations of the Information Technology Act.	07	
5	Business Organizations and E-Governance: Sole Traders, Partnerships: The Company's Act: Introduction, Formation of a Company, Memorandum of Association, Articles of Association, Prospectus, Shares, Directors, General Meetings and Proceedings, Auditor, Winding up. E-Governance and role of engineers in E-Governance, Need for reformed engineering serving at the Union and State level, Role of I.T. professionals in Judiciary, Problem of Alienation and Secessionism in few states creating hurdles in Industrial development.	05	
Course Outcome: After completion of this course, the learners will be able to			
CO1	Identify the basic features and modalities about Indian constitution.		
CO2	Differentiate the functioning of Indian parliamentary system at the center and state level.		
CO3	Differentiate different aspects of Indian Legal System and its related bodies.		
CO4	Apply different laws and regulations related to engineering practices.		
CO5	Correlate role of engineers with different organizations and governance model.		
Learning Resources:			
1.	Indian polity, M, Laxmikanth, MC Graw Hill education, 5th Edition.		
2.	Indian Constitution, M P Jain, 8 th Edition.		
3.	Indian Constitution and Administration, Latika Shekhar.		
4.	D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.		
5.	Dr. S. N. Busi, Dr. B. R. Ambedkar framing of Indian Constitution, 1st Edition, 2015. municipal Corporation.		

Name of the course		Computer Networking Lab	
Course Code: PCL(CS)520		Semester: 5 th	
Duration: 6 months		Maximum Marks: 100	
Teaching Scheme		Examination Scheme, Total Marks: 100	
Theory: Nil		Attendance : 10	
Tutorial: Nil		Preparation of Lab Report : 30	
Practical: 3 hrs./week		Experimental data/ Precision of work done : 30	
Credit Points: 1.5		Presentation/ analysis of the result : 10	
		Viva Voce: 20	
Module	Content	Hrs	Marks
1.	NIC Installation & Configuration (Windows/Linux)	01	
2.	Understanding IP address, subnet, MAC address, IP configuration	02	
3.	Networking cables (CAT5, UTP), Connectors (RJ45, T-connector)	01	
4.	Physical verification of existing LAN	01	
5.	5. TCP/UDP Socket Programming i) UDP time client server program ii) UDP echo client server program iii) TCP time client server program iv) TCP echo client server program v) TCP chat client server program Vi) Data Link Layer Error Detection Mechanism (Cyclic Redundancy Check)	12	
6.	Server Setup/Configuration FTP, Telnet, DNS.	06	
7.	Firewall configuration in client level	02	
8.	Mini project: Multiple user chat server implementation	02	
9.	Visualize different packet transfer methodology using packet tracer	03	
Course Outcome:			
After completion of this course the students will be able to -			
CO1	Investigate configuration of existing LAN		
CO2	Configure different components of computer network		
CO3	Implement client server model using socket programming		
CO4	Implement different server configuration		
CO5	Configure firewall		
CO6	Design of real life problems and solution for multiple client chat server		
Learning Resources:			
1	TCP/IP Sockets in Java, Practical Guide For Programmers Second Edition Kenneth L. Calvert and Michael J. Donahoo, The Morgan Kaufmann Practical Guides Series		
2	TCP/IP Sockets in C, Practical Guide For Programmers Second Edition Kenneth L. Calvert and Michael J. Donahoo, The Morgan Kaufmann		
3	A.S.Tanenbum – “Computer Networks(4 th Ed.)”- Pearson Education/PHI		
4	W.Stallings-“Data and computer communications(5 th Ed.)-PHI/Pearson Education		
5	Black ,Data & computer Communication, PHI		
6	Kurose and Rose-“ Computer networking –A top down approach featuring the internet”-Pearson Education		

Name of the course	Object Oriented Programming Lab
Course Code: PCL(CS)521	Semester: 5 th
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme, Total Marks: 100
Theory: Nil	Attendance : 10
Tutorial: Nil	Preparation of Lab Report : 30
Practical: 3 hrs./week	Experimental data/ Precision of work done : 30
Credit Points: 1.5	Presentation/ analysis of the result : 10
	Viva Voce: 20

Objective

1.	To understand fundamental Object-Oriented principles such as encapsulation, inheritance, and polymorphism in Java
2.	To analyze code reusability and execution safety based on underlying package architectures and exception boundaries
3.	To implement various multi-threaded synchronization models and interactive delegation event interfaces.
4.	To design data-centric software engines connecting applications to databases via JDBC and ORM Hibernate middleware frameworks.

Pre-Requisite: C programming knowledge

Module	Content	Hrs	Marks
1.	Construction & execution of java program with class, class data members, methods, constructors and using different access protection mechanism for those class members. Use of array of objects, method overloading, passing of object in method and returning of object from method, this and final keyword.	03	
2.	Java program writing on static variables & methods, call by value & call by reference scenarios, key board input operations with BufferedReader class, swing related I/O operations.	03	
3.	Java Program writing on different string handling related operations with String class properties, use of length(), equal(), charAt() methods. Java program writing on command line arguments	03	
4.	Java Program implementation for nested classes, name conflict resolving for inner and outer classes	03	
5.	Java Program writing and execution with method overriding, abstract class, interface, inheriting multiple interfaces in a single class, extending multiple interfaces, combined inheritance of both abstract class and interface. Use of polymorphism for abstract class and interface implementation.	03	
6.	Java Program implementation with nested abstract class and interface combinations. Resolving name conflict scenarios for the combined inheritance of abstract class and interface.	03	
7.	Designing java program modules with creation and accessing of packages.	03	
8.	Handling exception for java programs with try, catch and finally. Utilization of throw, throws and user defined exception concept.	03	
9.	Java Program writing for creation of multiple threads, thread synchronization and inter thread communication.	03	
10.	Java Program writing for basic file handling related operations, JDBC connectivity and data accessing	03	

Course Outcome:

After completion of this course the students will be able to -

CO1	Implement java program with class data protection, method overloading, object independent class member accessing scenarios, string handling operations and nested class members access features
CO2	Demonstrate inheritance, run time polymorphism and abstraction scenarios of java program in combination with java modular programming
CO3	Solve different exception cases in a java program that may evolve during execution
CO4	Design parallel processing scenarios with java multithread programming
CO5	Apply different data storage and retrieval mechanism with java application program and user query

Learning Resources:

1	Herbert Schildt: Java: The Complete Reference, (Eleventh/Twelfth Edition), McGraw Hill (TMH), 2021.
2	Timothy Budd: Understanding OOP with Java, Updated Edition, Pearson Education, 2004.