Curriculum for B.Tech. in Information Technology

(Applicable from the academic session 2024-2025)

Department of Information Technology

Government College of Engineering & Ceramic Technology

(An Autonomous Institution under MAKUT)

73, A.C Banerjee Lane

Kolkata-700010

Definition of Credit:

1 Hr. Lecture (L) per week	1 credit
1 Hr. Tutorial (T) per week	1 credit
1 Hr. Practical (P) per week	0.5 credits

UG COURSE STRUCTURE

1st SEMESTER							
Mandatory Induction Program- 3 Weeks duration							
SL. NO.	**TYPE OF COURSE**	**COURSE CODE**	**COURSE TITLE**	**HOURS PER WEEK**			**Credit**
				Lecture	**Tutorial**	**Practical**	
THEORY							
01	Basic Science course	BS(CS/IT) 101	Mathematics – I	3	0	0	3
02	Basic Science course	BS(CS/IT) 102	Physics	3	0	0	3
03	Engineering Science Course	ES(CS/IT) 101	Basic Electrical Engineering	3	0	0	3
04	Humanities & Social Sciences	HS(CS/IT) 101	English	2	0	0	2
SESSIONAL/PRACTICAL							
01	Basic Science course	BSL(CS/IT) 103	Physics Laboratory	0	0	2	1
02	Engineering Science Course	ESL(CS/IT) 102	Basic Electrical Engineering Laboratory	0	0	2	1
03	Engineering Science Course	ESL(CS/IT) 103	Engineering Graphics & Design	1	0	2	2
04	Humanities & Social Sciences	HSL(CS/IT) 102	English Communication Lab.	0	0	2	1
TOTAL				**12**	**0**	**8**	**16**

2nd SEMESTER							
SL. NO.	**TYPE OF COURSE**	**COURSE CODE**	**COURSE TITLE**	**HOURS PER WEEK**			**Credit**
				Lecture	**Tutorial**	**Practical**	
THEORY							
01	Basic Science course	BS(CS/IT) 204	Mathematics – II	3	0	0	3
02	Basic Science course	BS(CS/IT) 205	Chemistry	3	0	0	3
03	Engineering Science Course	ES(CS/IT) 204	Programming for Problem solving	3	0	0	3
04	Engineering Science Course	ES(CS/IT) 205	Analog and Digital Electronics	3	0	0	3
05	Humanities & Social Sciences including Management	HS(CS/IT) 203	Universal Human Values	2	0	0	2
SESSIONAL/PRACTICAL							
01	Basic Science course	BSL(CS/IT) 206	Chemistry Laboratory	0	0	3	1.5
02	Engineering Science Course	ESL(CS/IT) 206	Programming for Problem solving Laboratory	0	0	3	1.5
03	Engineering Science Course	ESL(CS/IT) 207	Workshop /Manufacturing Practices	1	0	2	2
04	Engineering Science Course	ESL(CS/IT) 208	Electronics Lab.	0	0	3	1.5
05	Humanities & Social Sciences including Management	HSL(CS/IT) 204	Design -Thinking Lab.	0	0	2	1
06	Community Service	ECA(CS/IT) 201	NSS				1
TOTAL				**15**	**0**	**13**	**22.5**

2 to 4 weeks training/internship in summer to be evaluated in 3rd Semester

3rd SEMESTER							
SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs./WEEK	CREDIT
THEORY							
01	BS(IT)307	Mathematics- III	3	0	0	3	3
02	ES(IT) 309	Communication Engineering	3	0	0	3	3
03	PC(IT)301	Computer Organization	3	1	0	4	4
04	PC(IT)302	Data structure & Algorithms	3	1	0	4	4
05	HS(IT)305	Economics for Engineers	3	0	0	3	3
06	BS(IT)308	Environmental Sciences	2	0	0	2	2
SESSIONAL/PRACTICAL							
01	PCL(IT)303	Computer Organization Lab	0	0	3	3	1.5
02	PCL(IT)304	Data structure & Algorithms Lab	0	0	3	3	1.5
03	PCL(IT)305	IT Workshop	0	0	3	3	1.5
04	ESL(IT)310	Communication Engineering Lab.	0	0	3	3	1.5
05	TRN(IT)301	Summer Internship 1	0	0	0	0	1
TOTAL			**17**	**2**	**12**	**31**	**26**

4th SEMESTER							
SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs./WEEK	CREDIT
THEORY							
01	BS(IT)409	Discrete Mathematics	3	0	0	3	3
02	PC(IT)406	Design & Analysis of Algorithm	3	1	0	4	4
03	PC(IT)407	Formal Language and Automata Theory	3	1	0	4	4
04	PC(IT)408	Computer Architecture	3	1	0	4	4
05	ES(IT)411	Biology for Engineers	3	0	0	3	3
06	HS (IT)406	Constitution of India	2	0	0	2	2
SESSIONAL/PRACTICAL							
01	PCL(IT)409	Algorithm Lab	0	0	3	3	1.5
02	PCL(IT)410	Programming Lab using C++	0	0	3	3	1.5
04	PROJ(IT)401	Project 1 (part1)	0	0	2	2	1
TOTAL			**17**	**3**	**8**	**28**	**24**

2 to 4 weeks training/internship in summer to be evaluated in 5th Semester)

5th SEMESTER							
SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs./WEEK	CREDIT
THEORY							
01	PC(IT)511	Operating Systems	3	1	0	4	4
02	PC(IT)512	Database Management System	3	1	0	4	4
03	PC(IT)513	Object Oriented Programming	3	1	0	4	4
04	PC(IT)514	Artificial Intelligence	3	0	0	3	3
05	OEC(IT)501	Open Elective-1	3	0	0	3	3
SESSIONAL/PRACTICAL							
01	PCL(IT)515	Operating System Lab	0	0	3	3	1.5
02	PCL(IT)516	Database Management System Lab	0	0	3	3	1.5
03	PCL(IT)517	Programming Lab using Java	0	0	3	3	1.5
04	OECL(IT)502 (A/B/C/D)	Open Elective-I Lab.	0	0	2	2	1
05	PROJ(IT)502	Project 1 (part 2)	0	0	2	2	1
06	TRN(IT)502	Summer Internship 2	0	0	0	0	1
TOTAL			**15**	**3**	**13**	**31**	**25.5**

6th SEMESTER							
SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs./WEEK	CREDIT
THEORY							
01	PC(IT)618	Computer Networks	3	1	0	4	4
02	PC(IT)619	Compiler Design	3	0	0	3	3
03	PC(IT)620	Software Engineering	3	0	0	3	3
04	PC(IT)621	Introduction to Cyber Security	3	0	0	3	3
05	PEC(IT)601	Elective-I	3	0	0	3	3
06	OEC(IT)603	Open Elective-II	3	0	0	3	3
SESSIONAL/PRACTICAL							
01	PCL(IT)622	Computer Network lab	0	0	3	3	1.5
02	PCL(IT)623	Software Engineering Lab.	0	0	3	3	1.5
03	PECL(IT)602 (A/B/C/D)	Elective-I Lab.	0	0	2	2	1
04	PROJ(IT)603	Project 1 (part 3)	0	0	4	4	2
TOTAL			**18**	**1**	**12**	**31**	**25**

Industrial training (4 to 6 weeks training in summer to be evaluated in 7th Semester)

7th SEMESTER							
SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs./WEEK	CREDIT
THEORY							
01	PEC(IT)703	Elective-II	3	0	0	3	3
02	PEC(IT)704	Elective-III	3	0	0	3	3
03	OEC(IT)704	Elective-IV	3	0	0	3	3
04	OEC(IT)705	Open Elective-III	3	0	0	3	3
05	HS(IT)707	Industrial Management	3	0	0	3	3
SESSIONAL/PRACTICAL							
01	PROJ(IT)704	Project 2/ Internship / Industrial Project	0	0	6	6	3
02	TRN(IT)703	Industrial Training	0	0	0	0	2
TOTAL			**15**	**0**	**6**	**21**	**20**

8th SEMESTER							
SL. NO.	PAPER CODE	PAPER NAME	L	T	P	CONTACT HRs./WEEK	CREDIT
THEORY							
SESSIONAL/PRACTICAL							
01	PROJ(IT)805	Project 3/ Internship / Industrial Project	0	0	10	10	5
02	CVV(IT)801	Comprehensive Viva Voce	0	0	0	0	2
TOTAL			**0**	**0**	**10**	**10**	**7**

Name of the course		**MATHEMATICS-III**	
Course Code: BS(IT) 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)		
Module	**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** Review of first Order Differential Equation, Second order linear differential equations with constant coefficients, D-operator method, method of variation of parameters, Cauchy-Euler equation.	7	
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	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 2	evaluate multiple integrals and apply the techniques to different physical problems.
CO 3	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 4	apply the basic concepts of graph theory to network analysis, data analytics and many other branches of computer science.
CO 5	analyze circuits using periodic signals and Fourier series.
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:	**Communication Engineering**		
Course Code: ES(IT)309	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.			
Module	**Content**	**Hrs.**	**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	10	
2.	Frequency Modulation(FM): Concept of FM, Direct & Indirect Method, Bandwidth calculation of FM, Demodulation of FM. Phase Modulation(PM) : Concept of PM ,	05	

	generation of PM from FM		
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, Companding, DPCM, Delta Modulation and Adaptive Delta Modulation, signal to quantisation noise ratio in PCM system. ASK, FSK, PSK, QPSK	1	
4.	Data Formatting: NRZ-Unipolar, NRZ-polar, NRZ-Bipolar, RZ-Bipolar, Manchester Coding Synchronous and Asynchronous Data Transmission .	03	
5.	Fundamental concepts of wireless communication, networks, types, advantages, challenges and spectrum used. Introduction to cellular concept Understand frequency reuse, handoff strategies, and capacity improvement techniques in cellular networks & its architecture.	08	
Course Outcome: 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 for analog to digital signal conversion.		
CO3	Compare various techniques for digital communication techniques.		
CO4	Compare different line coding techniques.		
CO5	Understand fundamental Concepts about wireless & cellular communications.		
CO6	Evaluate cellular System design.		
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 Haykin (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 and Analog and Digital Communication by Jerry D Gibson, Published by MacMillan.		
7.	Communication Systems (Analog and Digital) by Sanjay Sharma, Published by S.K.Kataria & Sons		

Name of the course	**COMPUTER ORGANIZATION**
Course Code:	PC(IT)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, flip teaching, Term paper: 20 Marks
Credit Points: 4	End Semester Exam: 50 Marks (100 marks converted to 50)

Objective:

1. To identify different processor architectures and their performance measurement parameters.
2. To develop the concept of instruction set of a processor.
3. To develop the concept of memory hierarchy
4. To develop the concept of I/O architecture like bus configuration, DMA, RAID
5. To design simple pipeline architecture

Prerequisite: Digital Electronics

Module	**Content**	**Lecture Hours**
1	Introduction: History of computing, von Neumann machine, Instruction and data, fixed-point and floating point numbers, errors, IEEE standards	3
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; bitpair algorithm; high performance arithmetic	9
3	Control unit design: Hardwired control, micro-programmed control design – micro-instruction formats, problem solving on control optimization	3
4	Memory subsystem: Registers, Memory hierarchy, memory interfacing, virtual memory, cache memory, memory replacement techniques, address mapping, content addressable memory (CAM), memory interleaving, real life problem solution	9
5	Peripherals: Basic properties, bus architectures, control and arbitration, interfacing of I/O devices, data transfer schemes –programmed I/O, memory mapped I/O, I/0 mapped I/O, DMA, mass storage, RAID	7
6	Pipelining: Pipelining, data path and instructions, speed up, CPI, latency; linear / non-linear pipeline–reservation table, MAL; super-pipelined and super-scalar processors,	6

	concept of memory parallelism	
Course Outcomes: After completion of the course the learners will be able to		
CO1	Analyse fixed-point and floating-point systems number representation	
CO2	Apply machine's instruction set architecture (ISA) including basic instruction fetch and execute cycles, instruction formats, control flow, and operand addressing modes	
CO3	Analyze the design and functioning of a machine's central processing unit (CPU), the data path components (ALU, register file) and the control unit.	
CO4	Design memory organization systems and compare in terms of efficiency	
CO5	Analyse basic input/output functioning including program controlled I/O and interrupt I/O	
CO6	Compare and contrast performance improvement of system using instruction and memory level parallelism	
Learning Resources:		
1	Mano, M.M., “Computer System Architecture”, PHI.	
2	Hayes J. P., Computer Architecture & Organisation, McGraw Hill	
3	Hamacher, Computer Organisation, McGraw Hill,	
4	Behrooz Parhami“ Computer Architecture”, Oxford University Press	
5	Tanenbaum, “Computer System Architecture”, PHI.	
6	Stalling “Computer Organization & Architecture, 11e”, Pearson	
7	Wang “Computer Architecture and Organization: Fundamentals and Architecture Security”, PHI	
8	Computer architecture and parallel processing; Kai Hwang &Faye A. Briggs, McGraw Hill	

Name of the course	**DATA STRUCTURE AND ALGORITHM**
Course Code: PC(IT)302	Semester: 3rd
Duration: 6 months	Maximum Marks: 100
Teaching Scheme	Examination Scheme
Theory: 3 hrs/week	Mid Term Exam I: 15 Marks
Tutorial: 1 hr./week	Mid Term Exam II: 15 Marks
Practical: NIl	Assignment.: 20 Marks
Credit Points: 4	Semester End Exam: 100 Marks (50% s weightage for final reckoning i.e., 50 marks)

Objective:			
1.	To Understand basic data structures such as arrays, linked lists and trees.		
2.	To Calculate the time complexities of accessing various data structures.		
3.	The ability to decide based on a given problem which data structure is appropriate.		
Pre-Requisite:			
1.	Programming for problem solving (ES(CS/IT) 204)		
Module	**Content**	**Lecture Hours**	
1	Introduction :Elementary Data Organizations, Data Structure Operations - insertion, deletion and traversal in arrays, asymptotic Notations, Time-Space trade off, recursion, tail recursion, Tower of Hanoi, recursion tree and master theorem method of complexity analysis, Linear Search and Binary Search Techniques and their complexity analysis, finding min max in O(3n/2) time.	10	
2	Stacks and Queues: ADT Stack and its operations; Algorithms and their complexity analysis, Applications of Stacks - Expression Conversion and evaluation – corresponding algorithms and complexity analysis; ADT queue and types of Queue- Simple Queue, Circular Queue, Operations on each type of Queue- Algorithms and their analysis.	8	
3	Linked List: Singly linked lists, Representation in memory, Algorithms of several operations -Traversing, Searching, Insertion into, Deletion from linked list; Linked List representation of Stack and Queue; Doubly linked list - operations, space and time analysis; Circular Linked Lists - all operations and complexity analysis; Floyd-Cycle finding algorithm.	8	
4	Trees: Basic Tree Terminologies, Different types of Trees - Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree, binary heap, B-tree; operations on each of the trees and their algorithms with complexity analysis; Tree traversal algorithms - recursive and iterative. Catalan Number and its connection to binary trees and stack sortable permutations ; Comparison of performance of Heap, array and insertion priority queues.	10	
5	Hashing: Chaining, probing, Universal hashing function and analysis of various hashing methods.	4	
Course Outcomes: After completion of this course, the learners will be able to-			
CO1	Analyze computation complexity of various algorithms.		
CO 2	Apply stacks and queues to various problems.		
CO 3	Implement linked list data structure		
CO 4	Analyze complexities of non linear data structures.		
CO 5	Compare performances of various hashing techniques		
Learning Resources:			
1	**Horowitz, Sahni, Anderson-Freed:** *Fundamentals of Data Structures in C* **(Second Edition), Universities Press, 2008.**		
2	**T.H. Cormen, C.E. Leiserson, R. Rivest and C. Stein:** *Introduction to Algorithms*,**(Second/Third Edition), PHI**, **2009.**		

3	**R. Sedgewick:** *Algorithms in C***, Pearson, 2004.**
4	Steven S Skiena, Algorithm design manual, 2nd Edition, Springer.
5	Steven S Skiena, Miguel A. Revilla, Programming Challenges: The Programming Contest Training Manual (Texts in Computer Science) Springer.

Name of the course	**ECONOMICS FOR ENGINEERS**
Course Code: HS(IT)305	**Semester: 3RD**
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: 100 Marks (50% 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.

Module	Content	Hours	Marks
1	• Introduction to Economics for Engineers – Basic Introduction to Economics, Productive resources, Scarcity and the Economic problem, Circular flow in an Economy, Production Possibility Frontier, Types of business organisation, Demand and Supply, Efficiency and sustainability, Engineering & Economics, Scope of Economics for Engineers, Fundamental principles of Engineering Economics, Role of Engineers in Economic Decision making, Problems 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, Explicit and Implicit cost, Out of pocket cost and Imputed cost, Recurring and Nonrecurring costs, Anticipated and Unanticipated costs, Differential or Incremental costs, Cash cost vs. Book 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. • 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.	9	
2	• 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.	4	

3	• Capital budgeting and Project selection – Basic concept of capital budgeting, Types of projects and cash flow patterns, features of a good capital budgeting criteria; Net Present Value (NPV) Analysis, NPV criteria for revenue dominated and cost dominated models, Internal Rate of Return (IRR) Analysis, Incremental Analysis, Comparison between NPV and IRR, Future Worth Analysis, Annual Worth Analysis, Payback period, Evaluation of Public Projects and Benefit-Cost Ratio Analysis, Sensitivity Analysis.	9	
4	• 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, Simulation, Real options analysis. • Inflation and Price Change – Definition, Effects, Causes and Stages of inflation, Price Change with Indexes, Types of Index, Tests of index numbers, Composite vs Commodity Indexes, Use of Price Indexes in Engineering Economic Analysis, Effect of inflation on project cash flows.	6	
5	• 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. • Introduction to Accounting – Basic concepts, scope, functions and limitations of Accounting, Financing business needs, Capital transactions, Cost accounting, Financial Statements, Financial Ratios, Uses and limitations of ratio analysis.	8	

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, Annual worth, Payback period, Benefit Cost Analysis and Sensitivity analysis.
CO 4	Identify the effects of risk, uncertainty and price change.
CO 5	Apply Depreciation, Replacement Analysis, Financial statements and Financial ratio 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 Eschembach, Jerome Lavelle : Engineering Economics Analysis, OUP
4.	Sullivan and Wicks: Engineering Economy, Pearson
5.	Chan S. Park, Contemporary Engineering Economics, Pearson
6.	Partha Chatterjee: Economics for Engineers, Vrinda Publications.

7.	James L. Riggs, David D. Bedworth, Sabah U. Randhawa : Economics for Engineers 4e , Tata McGraw-Hill .
8.	Dr. Shantanu Chakraborty, Nilanjan Singha Roy: : Economics for Engineers, Lawpoint Publications.
9.	N.G. Das, Statistical Methods, Tata Mcgraw Hill.

Name of the course		**ENVIRONMENTAL SCIENCES**	
Course Code: BS(IT)308		Semester: 3rd	
Duration: 6 month		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: 100 Marks (Fifty % 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 implimentations.		
Pre-Requisite			
Class 12 standard knowledge of physics, chemistry, biology, mathematics			
Module	Content	Hrs	Marks
1	. Green computing : Green IT Fundamentals, business ,IT and the environment ,Green IT strategies ; Drivers, Dimensions and Goals	6	
2	Green Assets and Modelling ; Buildings, Data Centres ,Networks and Devices , green enterprise architecture ,Environmental intelligence –green supply chains	6	
3	Carbon footprint : definition ,quantification, measurement .direct and indirect emissions, types , difference from Ecological Footprint .	5	
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	6	
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	7	

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	6	

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 ewaste 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	AnubhaKaushik, C.P. Kaushik, Perspectives in environmental studies, New Age International (P) Ltd, Publishers
2	ErachBharucha, 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 Ecololgy, 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	ShashiChawla, A Textbook of Environmental Studies, Tata McGraw Hill Education Private Limited.

Name of the course	**Computer Organization Lab**
Course Code: PCL(IT)303	**Semester: 3rd**
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

Module	**Content**
1.	Familiarization with IC chips: Multiplexer, Decoder, Priority Encoder, ROM, Comparator, Flip flop (Truth table verification and application)
2.	Design Adder, Subtractor using basic gates, Multiplexer and decoder
3.	Design Adder Subtractor composite unit
4.	Design BCD adder
5.	Design Carry look ahead adder circuit
6.	Design ALU(Arithmetic Logic Unit)
7.	Design of counter using Flip Flop
8.	Synthesize sequential circuits
9.	Execute Read and Write operation using RAM chip
10.	Cascading of RAM IC for vertical and horizontal expansion

Course Outcomes:
After completion of the course students will able to -

CO1	Asses different Integrated circuits
CO2	Design combinational circuits
CO3	Design sequential circuits
CO4	Implement different real life applications of combinational and sequential circuits required for basic computer architecture.
CO5	Evaluate different applications for higher order design

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(IT)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 Point:1.5	Presentation/ analysis of the result: 10
	Viva Voce:20

Objective:

1.	To understand the working of basic data structures..
2.	To analyse the performance of various data structures
3.	To implement various data structures
4.	To understand the difference between linear and non-linear data structure

Pre-Requisite:

1.	

Module	Content	Hours	Marks
1	Application of array insertion, deletion and traversal operations in solving problems.	03	
2	Linear Search, Binary Search Techniques and time complexity comparison.	03	
3	Application of binary search like divide and conquer technique in various array related O (log n) problems.	03	
4	Implementation and applications of Stacks and queues using arrays.	03	
5	Implementation of Singly linked lists, Linked representation of Stack and Queue.	03	
6	Implementation of Binary Search Tree.	03	
7	Application of binary trees in solving various problems.	03	
8	Array implementation of binary heap.	03	
9	Comparison of performance of binary Heap and array as priority queues.	03	
10	Implementation of B-Tree.	03	
11	Implementation of Chaining and probing techniques of collision resolution in hashing.	03	

Course Outcomes:

After completion of this course, the learners will be able to -

CO1	**Implement** linear data structures.
CO2	**Analyze** data sets and problems.
CO3	**Implement** non-linear data structures.
CO4	**Compare** various searching techniques.
CO5	**decide** which data structure to implement based on the problem.

Learning Resources:

1.	**Horowitz, Sahni, Anderson-Freed: *Fundamentals of Data Structures in C* (Second Edition), Universities Press, 2008.**
2.	**T.H. Cormen, C.E. Leiserson, R. Rivest and C. Stein: *Introduction to Algorithms*,(Second/Third Edition), PHI, 2009.**
3.	**R. Sedgewick: *Algorithms in C*, Pearson, 2004.**
4.	Steven S Skiena, Algorithm design manual, 2nd Edition, Springer.

Name of the course:	**IT WORKSHOP**
Course Code: PCL(IT)305	**Semester: 3rd**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory Contact Hrs.:	Attendance : 10
Tutorial Contact Hrs.:	Preparation of Lab Report : 20
Practical: 3 hrs./week	Experimental data/ Precision of work done : 30
Credit Point: 1.5	Presentation/ analysis of the result : 20
	Viva Voce: 20

Objective:	
1.	To implement Python programs using core Python programming concepts and functions
2.	To understand Object Oriented Python Programming techniques
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	10
2.	**Strings** Accessing Values in Strings, Traversing a String, String Operators, Built-In String Methods	3	10
3.	**Lists** Creating a List, Accessing Lists, Difference between String and List, Traversing a List, List Operations	3	10
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	10
5.	**Dictionary** 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	3	10
6.	**Introduction to Python Modules** Math Module, Random Module, Statistics Module	3	10
7.	**Functions** Scope, Parameter passing, Passing strings, Default parameters, Return values, Positional parameters	3	10

8.	**Object Oriented Programming(OOP) With Python** Basics of OOP, Class and Objects, Inheritance, Types of Inheritance	6	10
9.	**File Handling** Need for data file, Types of file :Text, Binary and Comma separated value files	3	10
10.	**Data Structures** Stacks : Push, Pop using a list, Queues : Insert, Delete using a list	3	10
Course Outcomes: After completion of this course the students will be able to -			
CO1	**Interpret** the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements		
CO2	**Express** proficiency in the handling of strings and functions		
CO3	**Identify** the commonly used operations involving file systems		
CO4	**Apply** object oriented programming concepts		
CO5	**Determine** the methods to create and manipulate Python programs by utilizing lists, tuples and dictionaries		
Learning Resources:			
1.	https://www.anaconda.com		
2.	Rakesh K. Yadav, Srinivas Arukonda, Monu Singh, Tapasya Dinkar, Dileep Kumar Yadav, Zero to Mastery in Python Programming, Vayu Education of India, ISBN: 9789389769364		
3.	Pooja Sharma, Programming in Python, BPB Publications, ISBN: 9789386551276		
4.	Reema Thareja, Python Programming- Using Problem Solving Approach, OUP India, ISBN: 9780199480173		

Name of the course		**Communication Engineering Lab.**	
Course Code: ESL(IT)310		**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 : 20	
Practical: 3 hrs./week		Experimental data/ Precision of work done : 30	
Credit Points: 1.5		Presentation/ analysis of the result : 20	
		Viva Voce: 20	
Module	**Content**	**Hours**	**Marks**
1.	Amplitude Modulation and Demodulation	03	
2.	Frequency modulation and Demodulation.	03	
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	03	
7.	Time Division Multiplexing & Demultiplexing	03	

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		**DISCRETE MATHEMATICS**	
Course Code: BS(IT)409		**Semester: 4th**	
Duration: 6 month		**Maximum Marks: 100**	
Teaching Scheme		**Examination Scheme**	
Theory: **3hrs./week**		Mid Term 1 Exam: 15 Marks	
Tutorial: 0hr./week		Mid Term 2 Exam: 15 Marks	
Credit Points: **3**		Other Assessment tools (Assignment, Quiz etc.): 20 Marks	
		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)		
Module	**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 (Zn) 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 (up to 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:** 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.	10	

5	**Module 5: Graph Theory:** Planar and Dual Graphs. Kuratowski's graphs. Homeomorphic graphs. Euler's 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. Umaparvathi, Discrete Mathematics, PHI.		
3.	J.K. Sharma, Discrete Mathematics, Macmillan.		
4.	Malik,Mordeson,Sen, Fundamentals of abstract algebra, Tata McGraw-Hill.		
5.	Clark John, Holton Derek Allan, A First Look at Graph Theory, World Scientific.		
6.	Kenneth H. Rosen, Discrete Mathematics and Its Applications, Sixth Edition, McGraw-Hill, 2006. ISBN 0073312711		

Name of the course	**DESIGN AND ANALYSIS OF ALGORITHMS**		
Course Code: PC(IT)406	**Semester: 4th**		
Duration: 6 months	**Maximum Marks: 100**		
Teaching Scheme	**Examination Scheme**		
Theory: **3 hrs./week**	Mid Semester Exam: 30 (15 + 15) Marks		
Tutorial: **1 hr./week**	Assignment & Quiz: 15 Marks		
Credit Points: **4**	Attendance: 05 Marks		
	End Semester Exam: 100 Marks (to be mapped into 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.	To decide based on a given problem which design paradigm and algorithm is appropriate.		
Pre-Requisite			
1.	Data Structure and Algorithms(PC(IT)302)		
Module	**Content**	**Hrs.**	**Marks**

1	**Models of computation & Algorithm design frameworks:** Models of computation - RAM model, Deterministic and Non-deterministic problems, Tractable and Intractable problems, Solvability, Algorithm design frameworks – Divide and Conquer, Backtracking, Greedy, Dynamic Programming, Decision and Optimization problems; Comparison - Divide & Conquer, Greedy and Dynamic Programming.	6	
2	**Sorting:** Comparison based sorts - Bubble sort, insertion sort, selection sort, quick sort, merge sort, analysis and comparison. Non-comparison based sorts - Radix sort, count sort; Median order statistics; Lower bound of sorting.	10	
3	**Illustrations of various design frameworks :** Dynamic Programming - Optimal substructure and overlapping sub problems; Matrix-chain multiplication; Backtracking - 8-queens problem; Greedy Method - Knapsack problem, Job sequencing with deadlines.	10	
4	**Graph Algorithms:** BFS and DFS- algorithm and comparison; Single source shortest path, All pair shortest paths; Prim's and Kruskal's algorithms for finding minimum spanning tree.	10	
5	**String matching problem:** Naive algorithm, Knuth-Morris-Pratt (KMP) algorithm.	3	
6	**Amortized Analysis:** Basic concept of amortized analysis, disjoint set data structure.	5	
7	**P and NP :** P, NP, NP-hard, NP-complete; reduction (concept only); Cook’s theorem (statement only)	4	
Learning Resources:			
1	T.H.Cormen, C.E. Leiserson, R.L.Rivest and C. Stein, “Introduction to Algorithms”, PHI.		
2	Ellis Horowitz, Sartaz 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 VenGelder“Computer Algorithms-Introduction to Design & Analysis"- 3rd Edition, Pearson Education		
Course Outcome: After completion of this course, the learners will be able to			
1	**Classify** algorithms as on the basis of various design paradigms.		
2	**Analyze** a problem to determine which design paradigm to use to solve the problem.		
3	**Distinguish** between problems employing divide and conquer, greedy and dynamic programming.		
4	**Solve** various graph problems efficiently.		
5	**Identify** whether a problem is in P or NP		

Name of the course	**FORMAL LANGUAGE AND AUTOMATA THEORY**
Course Code: PC(IT)407	**Semester: 4th**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: **3 hrs./week**	Mid Semester Exam: 30(15 + 15) Marks
Tutorial: **1 hr./week**	Assignment & Quiz: 15 Marks
Credit Points: **4**	Attendance: 05 Marks
	End Semester Exam: 100 Marks(to be mapped into 50 marks)
Objective:	

1.	To develop methods for describing and analysing the behaviour of discrete systems using abstract machines called automata.		
2.	To understand the limits of computation, how to build and verify systems.		
3.	To create foundational tools for designing programming languages and compilers.		
Pre-Requisite			
1.	Mathematics		
Module	**Content**	**Hrs.**	**Marks**
1.	**Introduction:** Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.	2	
2.	**Regular Language and Finite Automata:** Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages (proof not required), pumping lemma for regular languages, minimization of finite automata.	10	
3.	**Context-free languages and pushdown automata:** Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (NPDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs(proof not required).	9	
4.	**Context-sensitive languages:** Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.	4	
5.	**Turing Machines:** The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.	9	
6	**Undecidability:** Universal Turing machine, the universal and diagonalization languages, Rice s theorem.	2	
Course Outcome: After completion of this course the students will be able to -			
CO1	**understand** the class to which a language belongs.		
CO2	**design** finite automaton, grammar, expressions for regular languages.		
CO3	**design** pushdown automaton, grammar, for context free languages.		
CO4	**design** context-sensitive grammar and its languages, linear bound automata.		
CO5	**analyze** correctness of automata for various languages.		
CO6	**evaluate** whether a language is decidable or undecidable.		
Learning Resources:			
1.	Peter Linz, An Introduction to Formal Languages and Automata, Jones & Bartlett Learning.		
2.	Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.		

3.	John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.
4.	Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
5.	Michael Sipser, Introduction to the Theory of Computation, PWS Publishing. 4. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill

Name of the course	**COMPUTER ARCHITECTURE**
Course Code: PC(IT)408	**Semester: 4th**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: **3 hrs./week**	Mid Term I: 15 Marks
Tutorial: **1 hr./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 define processor performances for sequential and pipelined processors..
2.	To understand the function of pipeline processor.
3.	To understand cache and virtual memories.
4.	To develop the concept of multiprocessor architecture.
5.	To understand hardware description language for digital design.

Pre-Requisites :

1.	Digital Electronics [ES(CS/IT)307]
2.	Computer Organization [PC(CS/IT)301]

Module	Content	Hrs.	Marks
1	**Pipelining Architecture**: Introduction: Review of basic computer architecture (Revisited), Quantitative techniques in measuring computer performance. Pipelining: Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards and structural hazards, techniques for handling hazards, super pipelined and superscalar processor, exception handling.	8	25
2	**Memory Module:** Hierarchical memory technology, memory parameters, construction of semiconductor memory cell : SRAM, DRAM, read / write operations, IC chips for organization of RAMs, design of 2D memory.	4	10
3	**Cache and Virtual memory** : Coherence and locality properties; hit ratio, cache memory organizations, cache mapping techniques; Virtual memory organization, Address translation, paging and segmentation.	7	15
4	**Instruction-level parallelism**: Basic concepts: RISC Architecture, and VLIW processor architectures. Array and Vector processors.	6	15
5	**Multiprocessor Architecture**: Taxonomy of parallel architectures; Centralized shared-memory architecture: synchronization, memory consistency, interconnection networks. Cluster computers. Non von Neumann architectures: Data flow computers, Reduction computer architectures, Systolic architectures, Distributed Computing Architecture.	8	30
6	**Introduction to Computer aided design tools:** Introduction to Computer aided	5	5

	design tools for digital systems, Hardware description language: Introduction to VHDL.		
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	Design 2D memory organization.		
CO3	Analyze the performance of cache and virtual memory using different mapping techniques		
CO4	Analyze the concepts of Instruction level parallelism, VLIW architecture, Array and Vector processor		
CO5	Explain multiprocessor architecture, Cluster computers, Distributed Computing Architecture and Non von Neumann architectures.		
CO6	Design of basic digital circuits using computer aided design tools.		
Learning Resources:			
1.	Advanced Computer Architecture-Kai Hwang & Naresh Jotwani, 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.	A VHDL Primer - Jairam Bhaskar - P T R Prentice Hall		
6.	Computer Architecture and Organization - Hayes, McGraw-Hill.		
7.	Computer Organization-5th edition- Carl Hamacher, Zvonko, Vranesic, Safwat Zaky, McGraw Hill.		

Name of the course	**BIOLOGY FOR ENGINEERS**		
Course Code: ES(IT)411	**Semester: 4th**		
Duration: 6 months	**Maximum Marks: 100**		
Teaching Scheme	**Examination Scheme**		
Theory: **3 hrs./week**	Mid Term I: 15 Marks		
Tutorial: 0 hr./week	Mid Term II: 15 Marks		
Credit Points: **3**	Assignment, Test based on assignments, Surprise tests, Quizzes, Presentations, etc. : 20 Marks		
	End Semester Exam: 50 Marks		
Objective:			
1.	To understand the fundamental concepts of biology relevant to engineering		
2.	To explain biological systems from a molecular to a systemic level		
3.	To describe the applications of biology in various engineering disciplines		
4.	To apply biological principles in the design and innovation of engineering solutions		
5.	To foster interdisciplinary thinking combining biology and engineering		
Pre-Requisite			
1.	High School Biology		
Module	**Content**	**Hrs.**	**Marks**
1.	**Introduction to Engineering Biology** Relation of biological systems with engineering systems, Introduction to living	03	

	systems, Cell theory, Types of cells, Cell structure and function		
2.	**Biomolecules** Carbohydrates, amino acids, proteins, lipids, nucleic acids: Structure and functions, Enzymes: Types, function and kinetics	04	
3.	**Genetics and Molecular Biology** Molecular Components of Life, RNA and Types of RNA, DNA replication, transcription, translation, Genetic code and regulation of gene expression, Mendelian genetics and Patterns of Inheritance, genetic engineering basics	07	
4.	**Medical Imaging and Biological Systems** Introduction to Medical Imaging, Ultrasound Imaging (USG), X-Ray Imaging and CT-Scan, Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Integration: Multi-Modal Imaging and Case Studies	07	
5.	**Human Physiology** Foundations of Human Physiology, Cardiovascular and Respiratory Systems, Nervous System and Brain, Abdominal Organs and Digestive Physiology, Skeletal System and Musculoskeletal Physiology, Sensory Physiology and The Eye, Bioelectricity and biosignals, Introduction to biomedical devices	07	
6.	**Biotechnology, Synthetic Biology and Applications** Modeling biological processes, Tissue engineering, stem cells, Biosensors and Bioreactors, Applications in precision healthcare, smart agriculture and sustainable environment	08	

Course Outcome: After completion of this course the students will be able to -	
CO1	**understand** the basic structure and function of cells and biomolecules
CO2	**explain** genetics and the molecular mechanisms of heredity
CO3	**evaluate** the suitability of different imaging methods for specific clinical or biological applications
CO4	**analyze** different biological processes
CO5	**analyze** biological systems from an engineering perspective
CO6	**apply** concepts of biology in developing technologies in precision healthcare, smart agriculture and sustainable environment

Learning Resources:	
1.	Arthur T. Johnson, Biology for Engineers, CRC Press, ISBN: 9781420077636
2.	Neil A. Campbell, Biology, Pearson Education, ISBN: 9780321543257
3.	S. Srivastava, Molecular Biology and Biotechnology, CBS Publishers & Distributors, ISBN: 9788123904696
4.	William J. Thieman, Michael A. Palladino, Introduction to Biotechnology, Pearson Education Limited, ISBN: 9781292027616
5.	Lehninger, Nelson & Cox, Principles of Biochemistry, Macmillan Learning, ISBN: 9781464109621

Name of the course:	**CONSTITUTION OF INDIA**
Course Code: HS(IT)406	**Semester: 4th**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: **2 hrs./week**	Mid Semester-1 Exam: 15 Marks

Tutorial: 0 hr./week		Mid Semester-2 Exam: 15 Marks	
Credit Points: **2**		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.	Civics, Indian Constitution.		
Module	**Content**	**Hrs.**	**Marks**
1	**Introduction and Basic Information about Indian Constitution** Historical Background of the Constituent Assembly, Government of India Act of 1935 and Indian Independence Act of 1947,Enforcementof 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.	06	
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.	05	
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.	04	
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	05	

	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.		
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.	04	
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,8th 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		**Algorithm Lab.**	
Course Code: PCL(IT)409		**Semester: 4th**	
Duration: 6 months		**Maximum Marks: 100**	
Teaching Scheme		**Examination Scheme**	
Practical: **3 hrs./week**		Attendance 10	
Credit Point: **1.5**		Preparation of Lab Report: 30	
		Experimental data/Precision of work done: 30	
		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 algorithm to employ based on nature of problem.		
Pre-Requisite			
1.	Data Structure & Algorithms Lab PCL(IT)304		
Module	**Content**	**Hrs.**	**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.	03	
4	Solving 8 queens problem using backtracking and brute force method with comparison of performance.	03	
5	Solving of Knapsack and job sequencing problems using greedy approach.	03	
6	Implementation of BFS and DFS both recursive and non-recursive versions and their performance comparison.	03	
7	Implementation of Prim's algorithm and performance comparison based on different data structures used.	03	
8	Implementation of Dijsktra'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 KMP algorithm.	03	

Course Outcomes:	
After completion of this course, the learners will be able to	
CO1	**Compare** performance of various sorting algorithms
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	**Implement** string matching algorithms
Learning Resources:	
1	T.H. Cormen, C.E. Leiserson, R. Rivest and C. Stein: *Introduction to Algorithms*,(Second/Third Edition), PHI, 2009.
2	R. Sedgewick: Algorithms in C, Pearson, 2004.
3	Steven S Skiena, Algorithm design manual, 2nd Edition, Springer.

Name of the course:	**Programming Lab. Using C++**
Course Code: PCL(IT)410	**Semester: 4th**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Practical: **3 hrs./week**	Attendance: 10 marks
Credit Point: **1.5**	Preparation of Lab Report: 30 marks
	Experimental data/ precision of work: 30 marks
	Presentation / analysis of the result: 10 marks
	Viva voce: 20 marks

Objective:	
1.	To learn the syntax and semantics of the C++ programming language
2.	To learn how to write inline functions for efficiency and performance.
3.	To learn how to implement copy constructors and class member functions
4.	To learn how to design C++ classes for code reuse
5.	To understand how C++ improves C with object-oriented features
Pre-Requisite	
1.	C programming lab
2.	Data structure Lab

Module	Content	Hrs.	Marks
1	Introduction to the source code writing, compilation and execution process of C++ programme. Writing C++ Programme using I/O stream, command line arguments. Basic loop control, functions with CBV and CBR, identification of variables with scope resolution operator.	03	
2	Programme writing on classes, creation of objects, constructors and destructors, accessing members, array of objects, accessing of static members.	03	
3	Programme writing on function overloading, constructor overloading and default constructor, Object passing as function arguments and returning of objects from functions.	03	
4	Programme writing on friend functions, local classes., dynamic initialization of objects.	03	
5	Programme writing on copy constructor, operator overloading - binary and unary operators. operator overloading using friend functions.	03	
6.	Programme writing on derived classes, implementation of single inheritance, multilevel inheritance, hierarchical inheritance with constructor calling sequence.	06	
7.	Programme writing on multiple inheritances, constructor calling in derived classes, virtual base classes.	03	
8.	Programme writing on abstract classes, pointer to objects, this pointer, pointers to derivedclass.	06	
9.	Programme writing on virtual functions and run time polymorphism.	03	
10.	Programme writing on basic Class and Function templates.	03	
Course Outcome: After completion of this course the students will be able to -			
CO1	Define the concept of object oriented programming.		
CO2	Implement the concepts of loop, functions, array & pointers in C++.		
CO3	Analyze the concept of classes/objects, constructor and destructor.		
CO4	Apply the concept of inheritance in programming.		
CO5	Apply the concept of encapsulation in programming.		
CO6	Implement the concept of polymorphism in programming.		
Learning Resources:			
1.	The C++ Programming Language (4th edition) by Bjarne Stroustrup		
2.	C++ Primer 5th Edition		
3.	A Tour of C++ (C++ in –Depth Series) 1st Edition		
4.	The Design and Evolution of C++		