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 & Algorithm**
Course Code: PC(IT)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 analyse 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 Data Structures and Algorithm Analysis** Definition and classification of Data Structures, Primitive and Non-primitive data structures, Abstract Data Types (ADT) Definition and characteristics of algorithms, Algorithm specification, pseudocode, Time and space complexity,	4	
2	**Module 2: Linear Data Structures** **Arrays and Strings:** Memory Representation of Arrays and address calculations, One-dimensional, Two-dimensional and multi-dimensional arrays, Operations and complexity, 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 :** **Stack operations and applications:** Expression evaluation, Recursion, Tail recursion, Tower of Hanoi, recursion tree **Queue types:** simple, circular, priority queues, Double -ended queue(Deque), Application of Queue **Implementations** using arrays and linked lists	10	
3	**Module 3: Non-Linear Data Structures** **Trees:** Tree terminology and Properties, 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 Introduction to file organization and indexing -B-Tree **Heaps :** Min-heap and max-heap, Heap operations and applications.	11	

	Graphs: Graph terminology and representations (adjacency matrix/list), Breadth-First Search (BFS) and Depth-First Search (DFS)		
4	**Module 4: Searching and Sorting Algorithms** **Searching:** Linear search, Binary search, Time complexity **Sorting:** Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort, Heap sort, Time and space complexity	8	
5	**Module 5: Hashing** Hash functions and collision resolution (chaining, open addressing, Linear Probing)	3	

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. Cormen, C.E. Leiserson, 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 Skiena, Algorithm design manual, 2nd Edition, Springer.
7.	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	9	

	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.		
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: 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 Algorithm 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	Array Operations: Implement operations such as insertion, deletion, traversal, and searching on arrays. String Operations: Implement basic string operations such as concatenation, comparison, searching, and substring extraction.	3	
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 Infix to Postfix Conversion and 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, selection, quick, merge). Heap Operations: Implement min-heap and max-heap. Perform heap sort .	6	
Course outcomes After completion of the course, a student would be able to:			
CO 1	Implement and apply fundamental data structures		
CO 2	Analyse the performance of algorithms		
CO 3	Solve problems using appropriate data structures		
CO 4	Implement and debug algorithms		
CO 5	Apply data structures to practical applications		
CO 6	Demonstrate good programming practices		
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.	Cormen, Thomas H., Leiserson, 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	GeeksforGeeks: 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**	
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	3	10

	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	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++		

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**

OEC(IT)501
A: Control System
B: Microprocessor and Interfacing
C: Digital Signal Processing
D: Advanced Computer Architecture

Name of the course	**OPERATING SYSTEMS**
Course Code: PC(IT)511	**Semester: 5th**
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	Class performance & Attendance: 20 Marks
	End Semester Exam & Viva: 50 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.	Data Structures & Algorithms -PC(CS/IT)302
2.	Computer Architecture – PC(CS/IT)408

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: S**cheduling 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	Analyze different types of operating system.
CO2	Select different types of kernel 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 Niranjan G. Shivaratri, TMH
6	Operating System. - Dhamdhere: - TMH
7	An Introduction to Operating Systems- Dietel H. N- Addison Wesley.

Name of the course	**Database Management System**
Course Code: PC(IT)512	**Semester: 5th**
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	Assignments, Test based on assignments, Surprise tests, Quizzes, Presentations, Attendance etc.: 20 Marks
	End Semester Exam: 100 Marks (to be mapped into 50 marks)

Objective:	
1.	To understand the basic concepts and the applications of database systems.
2.	To learn the fundamentals of data models and to represent a database system using ER diagrams.
3.	To study SQL and relational database design.
4.	To understand the fundamental concepts of transaction processing, concurrency control techniques and recovery procedures.
5.	To understand the internal storage structures using different file and indexing techniques which will help in physical DB design.
Pre-Requisite:	
1.	Data structure & Algorithms PC(CS/IT)302

2.	Discrete Mathematics BS(CS/IT)408		
Module	**Content**	**Hrs.**	**Marks**
1	**Introduction:** Concept of File system & Database system & their differences, Data abstraction & Data independence in DBMS, Instances & Schemas, Data models	2	
2	**Entity Relationship Model:** Basic concepts, Types of attributes, Relationship sets, Mapping cardinalities & Participation constraints, Types of Keys., Entity-Relationship diagram (E-R diagram), Strong & Weak entity sets, Specialization & Generalization & Aggregation in ER model	3	
3	**Relational Model and SQL:** Fundamental operations in Relational Algebra, Extended Relational Algebra operations, Concept of View, Relational Calculus, Characteristic of SQL, SQL operators & their procedures, Queries, Sub-queries & nested queries, Aggregate Functions, Operations on Modification of databases	8	
4	**Integrity Constraints and Normalization:** Concept of Foreign Key, Definition of integrity constraints, Types of integrity constraints (Domain Constraints, Entity Integrity Constraint, Referential Integrity Constraints, Key Constraints), Functional Dependency, Closure of functional dependency, Armstrong's Axioms, Canonical Cover, Lossless join decomposition & Dependency preservation, Full & Partial & Transitive dependency, Prime & Non-prime attribute, Need of Normalization, Database Normal Forms (1NF–5NF, including BCNF)	7	
5	**Transaction Management and Concurrency Control:** Overview of Database transaction concepts, ACID properties, Transaction state, Concurrent executions, Conflicts in Transaction, Serializability, Conflict & View Serializability, Test for serializability (Precedence Graph), Recoverability, Recoverable, Cascade less & Strict schedules, Shared lock & Exclusive lock, Two phase locking protocol, Deadlock handling, Causes of transaction failure, Storage structure, Log-based recovery, Write Ahead Logging (WAL) protocol, Timestamp Ordering Protocol, Checkpoints, Shadow paging.	12	
6	**Storage:** Single level and Multi level indexing, Structure of B and B^+ tree, File organization in B^+ tree, Hashing techniques.	4	
Course Outcomes: After completing the course, the student will be able to-			
CO1	Explain the concept of Database system.		
CO2	Design ER-models to represent simple database application scenarios.		
CO3	Implement Relational algebra and SQL queries on database.		
CO4	Apply integrity constraints and normalization for good database design.		
CO5	Solve concurrency problems in database transactions.		
CO6	Explain basic database storage structures and access techniques.		
Learning Resources:			

1.	Henry F. Korth and Silberschatz Abraham, "Database System Concepts", McGraw Hill, ISBN: 9780072283631
2.	Elmasri Ramez and Navathe Shamkant, "Fundamentals of Database Systems", Pearson Education India, ISBN: 9788131716250
3.	Ramakrishnan and Gehrke, "Database Management Systems", McGraw-Hill, ISBN: 9780071231510
4.	Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle",4th edition, BPB Publications ISBN: 9788176569644
5.	C.J. Date, "An Introduction to Database Systems" ,7th edition, Pearson,ISBN:9780321197849

Name of the course	**Object Oriented Programming**		
Course Code: PC(IT)513	**Semester: 5st**		
Duration: 6 months	**Maximum Marks: 100**		
Teaching Scheme	**Examination Scheme**		
Theory: 3	Mid Semester Exams: 30 Marks(2test of 15 marks)		
Tutorial: 1	Assignment, Quiz, Flip Teaching.: 20 Marks		
Credit Points: 4	End Semester Exam: 100 Marks (to be mapped into 50 marks)		
Objective:			
1.	To construct models for object-oriented software development		
2.	To inspect different run time exception cases in a java programme		
3.	To comprehend and write java programmes with abstraction, code reusability and data security features		
4.	To plan concurrent processing scenarios with java multithread programming.		
Pre-Requisite:			
1.	Programming for problem solving(ES(CS/IT)204)		
Module	**Content**	**Hrs.**	**Marks**
1	**Introduction to Object Oriented Programming Concepts** Object Oriented Programming language concepts & features, Comparison between Object Oriented Programming language and conventional programming languages, Object Oriented Modelling concepts.	2	
2	**Introductory Concept of Java Programming** Advantages of Java, Data types & variables, Loops, Arrays, Operators, Control statements, constants, methods, Compile time Polymorphism: Method Overloading, Keyboard input operations. Classes & Objects-Defining Classes and Creation of objects, Access specifiers, Instance variables and Static variables, Constructors, Constructor overloading, Static blocks, Array of objects, Use of this keyword, Passing objects as parameter to a method & returning objects from a method, Nested classes & Inner classes concept of string object with length(), equals() and charAt() method of string object, Command Line Arguments, garbage collection.	10	

3	**Inheritance and Polymorphism in Java** Concept of Inheritance, Super classes & Subclasses, Object Modelling in Java: Generalization and Specialization, Constructor calling mechanism in inheritance, Use of **super** keyword, Runtime Polymorphism: Method Overriding. Use of **static** keyword in java. Abstract classes & Interfaces-Concept of Abstract classes & Interfaces and their properties, use of final keyword, Dynamic binding in abstract classes and interfaces, Inheritance of interfaces, Nested Abstract classes & Nested Interfaces. Packages in Java-Creation of packages, Importing packages, Member access rules in the aspect of packages.	10	
4	**Exception handling in Java** Basic concept of exception handling in Java, Different types of exception classes, Concept of try and catch block, Concept of nested try block and multiple catch blocks, throw and throws clause, Concept of finally block, Creation of user defined exception classes.	5	
5	**Multithreading in Java** Basic concept of multithreading, Concept of main thread and child thread, Thread life cycle, Creation of multiple threads, Thread priorities, Thread synchronization, Inter thread communication, Deadlocks, Suspending & Resuming threads.	6	
6	**Applet Programming in Java** Basics of applet programming, Applet life cycle, Differences between application & applet programming, Parameter passing through applets, I/O operations in applets.	3	

Learning Resources:	
1	Cay S Horstmann and Gary Cornell , Core Java Volume I , Fundamentals (9th Edition), Prentice Hall
2	Rambaugh, James Michael, Blaha, Object Oriented Modelling and Design, Prentice Hall, India
3	Herbert Schildt, Java: A Beginner's Guide, Oracle Press.
4	Kathy Sierra and Bert Bates , Head First Java , O'reilly
5	Deitel and Deitel, Java How to Program", Pearson Education.

Course Outcomes: After completion of this course the students will be able to -	
CO1	Identify Object oriented programming features associated with object oriented modelling concepts related to object-oriented software development.
CO2	Apply various abstraction and code reusability features of java for more efficient and secure coding along with dynamic resolving of polymorphic behaviours of the entity in combination with java modular programming
CO3	Implement inheritance, run time polymorphism and abstraction features of java in combination with java modular programming
CO4	Examine different run time or compile time exceptional cases that may occur in a java program.
CO5	Organize different parallel processing scenarios with java multithread programming and make use of them in web applications through java applet programming

Name of the course	**ARTIFICIAL INTELLIGENCE**
Course Code: PC(IT)514	**Semester: 5th**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs./week	Mid Term I: 15 Marks
Credit Points: 4	Mid Term II: 15 Marks
	Assignments, Test based on assignments, Surprise tests, Quizzes, Presentations, Attendance etc.: 20 Marks
	End Semester Exam: 100 Marks (to be mapped into 50 marks)

Objective:

1.	The main purpose of this course is to provide the fundamental knowledge to the students so that they can understand what AI is.
2.	Apply the basic principles, models, and algorithms of AI to recognize, model, and solve problems in the analysis and design of information systems.
3.	Analyze the structures and algorithms of a selection of techniques related to searching, reasoning, Machine learning, and language processing.

Pre-Requisite

1.	Mathematics. (BS (CS/IT) 101, BS (CS/IT) 205, BS(CS/IT)307).
2.	Design & Analysis of Algorithm PC(CS/IT)406.
3.	IT Workshop(python) PCL(CS/IT)305.

Module	Content	Hrs	Marks
1	**Introduction to AI:** Overview of AI, AI tools and techniques, AI Platforms. Structure of agents, Goal based agents, Utility based agents, Learning agents.	3	
2	**Search techniques:** Problem Space & Search, Defining the problem as state space search, production system, problem characteristics, issues in the design of search algorithms. Searching for solutions; uniform search strategies: breadth first search, depth first search, depth limited search, bidirectional search, comparing uniform search strategies. Greedy best-first search, A* search, AO* search, Memory bounded heuristic search: local search algorithms & optimization problems, Hill climbing, Simulated annealing, Constraint satisfaction problems, Local search for constraint satisfaction problems, Adversarial search. Games, Optimal decisions & strategies in	11	

	games, Minimax search procedure, Alpha-beta pruning.		
3	**Knowledge & Reasoning:** Knowledge & Reasoning, Knowledge acquisition & representation, Knowledge representation, Frames and Scripts, Graphical representations, Resolution in Predicate Logic, Probabilistic reasoning, Procedural versus Declarative knowledge, Forward versus backward reasoning, Probabilistic reasoning.	6	
4	**Machine Learning and Planning:** Introduction, Machine Learning Process: Supervised, unsupervised, and reinforcement learning, Overview of planning, components of a planning system, Concepts of Deep Learning and Agentic learning.	8	
5.	**AI Tools & Applications:** Generative AI, Coding with Python, GitHub Copilot. AI in healthcare, Medical imaging, AI in finance, Risk assessment, AI in education: Personalized learning and intelligent tutoring systems.	8	

Course Outcome:

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

CO1	Explain the fundamental knowledge of AI, and the concepts of agents.
CO2	Analyze problem formulation in AI using state-space representation, production systems, and problem characteristics.
CO3	Analyze various knowledge representation schemes such as frames, scripts, and graphical representations and use of predicate logic.
CO4	Explain the fundamental concepts, objectives, and applications of Machine Learning and AI planning systems.
CO5	Explain the applications of modern AI tools, including Generative AI and the applications in healthcare, Finance and Education.

Learning Resources:

1.	Stuart Russell and Peter Norvig , Artificial Intelligence: A Modern Approach , Pearson
2.	George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson
3.	Elian Greystone, Mastering AI with Python: A Beginner's Guide to Machine Learning, Deep Learning, Generative AI, LLMs, and AI Agents, VertixCloud
4.	*Aurélien, Hands-On Machine Learning with Scikit-Learn, Keras and Tensor Flow, O'Reilly Media*
5.	Giuseppe Bonaccorso ,Armando Fandango and Rajalingappaa Shanmugamani, Advanced Guide to Artificial Intelligence: Expert machine learning systems and intelligent agents using Python, Packt Publishing

Name of the course	**Control System**
Course Code: OEC(IT)501A	**Semester: 7th**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs/week	Internal Assessment (50 Marks) a) Mid Semester Exam I: 15 Marks b) Mid Semester Exam II: 15 Marks c) Other Assessment tools (Assignment, Quiz etc.): 20 Marks
	End Semester Exam: 100 Marks (Converted to 50)
Credit Points: 3	

Objective:	
1.	To classify different systems and the related parameters.
2.	To apply different mathematical tools & techniques for analyzing different practical systems.
3.	To develop the concept of stability of a system and compute stability parameters.
4.	To design different controller parameters for stabilizing specific systems
Pre-Requisite	
1.	Basic Electrical Engineering (ES (CS/IT) 101)
2.	Mathematics (BS (CS/IT) 101, BS (CS/IT) 205, BS(CS/IT)307)

Unit	Content	Hrs	Marks
1	**Introduction to Control System:** Introduction to control system, objectives and areas of applications, Open loop system and closed loop system, Feedback control and Automatic control: concepts and examples, Concept and examples of linear and nonlinear systems, sensitivity, robustness, accuracy	3	
2	**Concept of transfer function: mathematical modeling of physical systems:** Transfer function of real life systems, properties and applications, Basic concepts of poles and zeroes of a transfer function, Mathematical modeling: electrical analogy of spring–mass-dashpot system, Block diagram representation of physical systems and analysis of block diagram, Different techniques for block diagram reduction, Development of signal flow graph, Mason's gain formula	7	
3	**Control system components:** Potentiometer, Synchros, Resolvers, Position encoders, Tacho-generators, Actuators, Basic concept of position control, speed control, temperature control, liquid level control, pressure control.	3	
4	**Time domain analysis:** Impulse, step and ramp function, Step response of first and second order system, Time domain analysis of a standard second order closed loop system, Understanding of Steady state error, undamped natural frequency, damping, overshoot, rise time and settling time and their applications, Stability assessment using locations of poles and zeroes, Stability analysis using Routh-Hurwitz criteria	7	
5	**Stability Analysis and control:** Stability analysis using Root locus techniques	11	

	from transfer function, Idea of semi-log graph, Bode plots and stability analysis using Bode plots from transfer function, Measurement of phase margin and gain margin, Development of polar plots from transfer function, Measure of relative stability using Nyquist criteria, PI, PD and PID control		
6	**Introduction to State variable Analysis:** State variables and state space model, Diagonalization, Solution of state equations, Computation of stability, controllability and observability from state model	5	

Course Outcome:

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

1. **Develop** transfer function of different systems using mathematical analysis, block diagram reduction, Mason's gain formula etc.
2. **Explain** the operation of different components of control system and physical control systems
3. **Examine** the system performance using different parameters of time domain response
4. **Determine** stability of a system using Root locus techniques, Bode plots and Nyquist criteria using transfer function of a system
5. **Measure** controllability and observability of a system from its state space model

Learning Resources:

1. Modern Control Engineering, K. Ogata, 5th Edition, Pearson Education India
2. Control System Engineering, I. J. Nagrath & M. Gopal. 6th Edition, New Age International Publication.
3. Automatic Control Systems, B.C. Kuo & F. Golnaraghi, 10th Edition, McGraw Hill India
4. Automatic Control Systems (with Matlab Programs), S. Hasan Saeed, Kataria, S. K., & Sons
5. Modern Control Engineering, D. Roy Choudhury, PHI Learning
6. Control Systems, A. Anand Kumar, 2nd Edition, PHI Learning
7. Linear Control Systems with MATLAB Applications, B. S. Manke, Khanna Publishers

Name of the course	**MICROPROCESSOR AND INTERFACING**
Course Code: OEC(IT) 501B	**Semester: 5th**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs./week	**Mid Term I: 15 Marks**
Credit Points: 3	**Mid Term II: 15 Marks**
	Assignment, Test based on assignments, Surprise tests, Quizzes, Presentations, etc. : 20 Marks
	End Semester Exam: 50 Marks
Objectives :	

1.	To understand the **architecture of 8085 and 8086 microprocessors.**
2.	To implement different techniques of assembly language programming.
3.	To develop the concept of hardware interfacing of peripheral devices.

Pre-Requisites :

1.	Analog and Digital Electronics [ES(CS/IT) 205]
2.	Computer Organization [PC(IT)301]
3.	Computer Architecture [PC(IT)408]

Unit	Content	Hrs	Marks
1	Introduction to Microcomputer based system. History of evolution of Microprocessor and Introduction to Microcontrollers and their advantages and disadvantages. Architecture of 8085 Microprocessor, Pin description of 8085. Address/data bus Demultiplexing , Status Signals and the control signals. Instruction set of 8085 microprocessor, Addressing modes, Timing diagram of the instructions (a few examples).	10	
2	Assembly language programming with examples, Counter and Time Delays, Stack and Subroutine, Interrupts of 8085 processor (software and hardware), I/O Device Interfacing-I/O Mapped I/O and Memory Mapped I/O, Serial (using SID and SOD pins and RIM, SIM Instructions) and Parallel data transfer.	10	
3	The 8086 microprocessor- Architecture, Addressing modes, Interrupts processors.	6	
4	Memory interfacing with 8085, Supporting IC chips- 8255, 8251,8237/8257,8259 8255 PPI with 8085	12	

Course Outcomes:

After completion of this course students will be able to-

CO1	Identify the internal architecture and pin diagrams of 8085 and 8086 microprocessors.
CO2	Explain the internal register organizations, buses and execution units of standard microprocessors like Intel 8085, 8086.
CO3	Write assembly language programs using specific instruction sets, addressing modes, and timing diagrams.
CO4	Analyze the concept of interfacing with I/O mapped I/O and memory mapped I/O.
CO5	Design interfacing of external devices with specific programmable peripheral interfaces like 8255 PPI.

Learning Resources:

1.	Microprocessors and microcontrollers - N. Senthil Kumar, M. Saravanan and Jeevananthan (Oxford university press)
2.	Fundamentals of Microprocessors and Microcomputers – B. Ram (Dhanpat Rai)

3.	Microprocessor architecture, programming and Application with 8085 - R.Gaonkar (Penram international Publishing LTD.)
4.	An Introduction to Microprocessor and Applications –Krishna Kant (Macmillan)
5.	Advanced Microprocessors and Peripherals : Architecture, Programming and Interfacing – A. K. Ray and K. M. Bhurchandi (TMH)

Name of the course	**DIGITAL SIGNAL PROCESSING**	
Course Code: OEC(IT)501C	**Semester: 5th**	
Duration: 6 months	**Maximum Marks: 100**	
Teaching Scheme	**Examination Scheme**	
Theory: 3 hrs./week	**Mid Term I: 15 Marks**	
Credit Points: 3	**Mid Term II: 15 Marks**	
	Assignment, Test based on assignments, Surprise tests, Quizzes, Presentations, etc. : 20 Marks	
	End Semester Exam: 50 Marks	
Course Objectives		
1.	Understand the mathematical foundations of discrete-time signals and systems.	
2.	Analyze discrete-time systems using convolution, Z-transform, and Fourier techniques.	
3.	Apply frequency-domain methods for signal representation and spectral analysis.	
4.	Implement computationally efficient algorithms for Fourier analysis.	
5.	Design and analyze digital filters for practical engineering applications.	
6.	Utilize DSP techniques in communication systems, multimedia processing, biomedical engineering, artificial intelligence, and data analytics.	
Pre-Requisite		
1.	Signals and Systems	
2.	Communication Engineering	
3.	Engineering Mathematics	
4.	Basic Programming Knowledge	
Module	***Content***	***Hours***

1	**Discrete-Time Signals and Sampling** Introduction to Digital Signal Processing, Continuous-Time and Discrete-Time Signals, Classification of Signals, Energy and Power Signals, Elementary Sequences (Unit Sample, Unit Step, Unit Ramp, Exponential and Sinusoidal Sequences), Arithmetic Operations on Sequences, Sampling Process, Sampling Theorem, Aliasing Effect and Signal Reconstruction.	4
2	**Discrete-Time LTI Systems** Discrete-Time Linear Time-Invariant (LTI) Systems, Impulse Response, Difference Equations, Input-Output Relationship, Linear and Circular Convolution, Graphical and Analytical Methods, Properties of Convolution, Stability and Causality Conditions.	6
3	**Z-Transform and System Analysis** Definition of Z-Transform, One-Sided and Two-Sided Z-Transform, Region of Convergence (ROC), Properties of Z-Transform, Inverse Z-Transform (Power Series, Long Division and Partial Fraction Methods), Pole-Zero Representation, Mapping Between s-Plane and z-Plane, Unit Circle, System Function H(z)H(z)H(z), Frequency Response from Z-Transform, Initial Value Theorem, Final Value Theorem, Stability and Causality Analysis Using ROC.	8
4	**Fourier Analysis of Discrete-Time Signals** Discrete-Time Fourier Transform (DTFT): Definition, Properties, Frequency Response of LTI Systems, Parseval's Relation, Relationship between DTFT and Z-Transform. Discrete Fourier Transform (DFT): Definition, DFT/IDFT Relations, Twiddle Factors, Matrix Representation, Circular Convolution, Linear Convolution Using DFT, Spectral Analysis. *(DTFS has been omitted to accommodate the reduced contact hours.)*	8
5	**Fast Fourier Transform (FFT)** Computational Complexity of DFT, Need for FFT, Radix-2 Decimation-in-Time (DIT) FFT, Radix-2 Decimation-in-Frequency (DIF) FFT, Butterfly Structure, Bit-Reversal Technique, Inverse FFT (IFFT), Comparison of DFT and FFT.	4
6	**Digital Filter Design** Introduction to Digital Filters, FIR and IIR Filters, Characteristics of Ideal Filters, FIR Filter Design Using Window Method, IIR Filter Design Using Impulse Invariant and Bilinear Transformation Methods, Butterworth Filter Design, Comparison of FIR and IIR Filters, Applications.	6

Course Outcomes:

After completion of this course students will be able to-

CO1	Analyze and classify discrete-time signals and systems and explain sampling phenomena.
CO2	Evaluate responses of LTI systems using convolution and difference equations.
CO3	Apply Z-transform techniques for system characterization, stability, and frequency-domain analysis.
CO4	Analyze discrete-time signals using DTFS, DTFT, and DFT representations.
CO5	Implement and compare FFT algorithms for efficient computation of spectral components.
CO6	Design and analyze FIR and IIR digital filters to satisfy prescribed specifications.

Learning Resources:

Text Books

1. Proakis, J. G., & Manolakis, D. G., *Digital Signal Processing: Principles, Algorithms and Applications*, Pearson Education.
2. Oppenheim, A. V., Schafer, R. W., & Buck, J. R., *Discrete-Time Signal Processing*, Pearson Education.
3. Mitra, S. K., *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill Education.
4. Salivahanan, S., Vallavaraj, A., & Gnanapriya, C., *Digital Signal Processing*, McGraw-Hill Education.
5. Ifeachor, E. C., & Jervis, B. W., *Digital Signal Processing: A Practical Approach*, Pearson Education.

Reference Books

1. Lyons, R. G., *Understanding Digital Signal Processing*, Pearson Education.
2. Rabiner, L. R., & Gold, B., *Theory and Application of Digital Signal Processing*, Prentice Hall.
3. Ingle, V. K., & Proakis, J. G., *Digital Signal Processing Using MATLAB*, Cengage Learning.

Online Resources

1. NPTEL Courses on Digital Signal Processing.
2. MATLAB DSP Toolbox Documentation.
3. Python Scientific Libraries (NumPy, SciPy, Matplotlib).
4. Virtual Labs – Digital Signal Processing.

Name of the course	**Operating System Lab**
Course Code: PCL(IT)515	**Semester: 5th**
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 of Linux Commands.
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).
3.	Implementation of CPU scheduling algorithms.
4.	Implementation of classical problems in process synchronization.
5.	Implementation of deadlock handling techniques.

6.	Implementation of memory management techniques.
7.	Operations on Processes, signals, Pipes and system calls.
Course Outcome: **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.
Learning Resources:	
1	Linux Command Line and Shell Scripting Bible- Christine Bresnahan and Richard BLUM- Wiley India
2	Linux Administration: The Linux Operating System and Command Line Guide- Jason Cannon- CreateSpace Independent Publishing Platform
3	Mastering Linux Administration- Alexandru Calcatinge, Julian Balog-- Packt

Name of the course	**DATABASE MANAGEMENT SYSTEM LAB**		
Course Code: PCL(IT)516	**Semester: 5th**		
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.	Describe the basics of SQL		
2.	Construct queries using SQL		
3.	Demonstrate the use of constraints		
4.	Implement PL/SQL Concepts and Constructs		
Pre-Requisite			
1.	Programming for Problem Solving Laboratory ESL(CS/IT)205		
2.	Discrete Mathematics BS(CS/IT)408		
Module	**Content**	**Hrs.**	**Marks**
1.	Structured Query Language: Creating a Database, Creating a Table, Specifying Relational Data Types, Specifying Constraints, Creating Indexes	03	
2.	Table and Record Handling: INSERT statement, INSERT INTO SELECT	06	

	statement, DELETE, UPDATE, TRUNCATE statements, DROP, ALTER statements		
3.	Retrieving Data from a Database: The SELECT statement, Using the WHERE clause, Using Logical Operators in the WHERE clause, Using IN, BETWEEN, LIKE, ORDER BY, GROUP BY and HAVING Clause, Using Aggregate Functions, Combining Tables using JOINS, Sub queries	09	
4.	Database Management: Creating Views, Creating Column Aliases, Creating Database Users, Using GRANT and REVOKE	06	
5.	PL/SQL Concepts and Constructs: Introduction Of PL/SQL, Structure of basic PL/SQL Structure, Conditional statements, Basic loops, Cursors in Oracle PL/SQL	06	

Course Outcome:

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

CO1	**Construct** Databases and Tables
CO2	**Manipulate** Tables and Records
CO3	**Compose** queries to retrieve data from a Database
CO4	**Facilitate** the management of a Database
CO5	**Implement** conditional statements, basic loops and cursors in PL/SQL

Learning Resources:

1	Ivan Bayross, SQL, PL/SQL the Programming Language of Oracle, BPB Publications, ISBN: 9788176569644
2	Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, McGraw-Hill, ISBN: 9789332901384

Name of the course	**Programming Lab Using Java**
Course Code: PCL(IT)517	**Semester: 5st**
Duration: 6 months	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Practical: 3 hrs/week	Assignments+ Final Test: 100 Marks (Internal: 80+ External: 20)
Credit Points: 1.5	

Objective:

1.	To construct models for object-oriented software development
2.	To handle different run time exception cases in a java programme
3.	To write java programmes with abstraction, code reusability and data security features
4.	To plan concurrent processing scenarios with java multithread programming.

Pre-Requisite:

1.			
Module	**Content**	**Hrs.**	**Marks**

1	Programming with java classes involving data members having various access protection, class methods, constructors, overloading features, this and final keyword, static block, static variables and methods.		
2	Use of array of objects, passing of object in method and returning of object form method, use of string handling functions– length (), equals (), charAt(), keyboard input operations, command line arguments.		
3	Program implementation for nested/inner classes, name conflict resolving for inner and outer classes.		
4	Program implementation for abstract class, interface, inheriting multiple interfaces in a single class, extending multiple interfaces within a single interface, combined inheritance of both abstract class and interface. Use of dynamic method dispatch for abstract class and interface implementation.		
5	Implementation of nested abstract class and interface combinations. Resolving name conflict scenarios for the combined inheritance of abstract class and interface.		
6	Designing program modules with creation and accessing of packages.		
7	Handling exception with try, catch and finally. Adoption of throw, throws and user defined exception.		
8	Program writing for creation of multiple threads, thread synchronization, inter thread communication.		
9	Applet program execution with I/O operation, use of repaint () method.		

Learning Resources:

1	Cay S Horstmann and Gary Cornell Core Java Volume I — Fundamentals (9th Edition), Prentice Hall
2	Harvey Deitel and Paul Deitel, Java How to Program, Early Objects, Global Edition, Pearson Education, ISBN-13: 9781292223902.
3	Herbert Schildt , Java: A Beginner's Guide, Oracle Press.
4	Kathy Sierra and Bert Bates, Head First Java, O'reilly.
5	Deitel and Deitel- "Java How to Program", Pearson Education.

Course Outcomes:

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

CO1	Implement java programs with data protection, method overloading, object independent class member accessing features and string handling operations.
CO2	Demonstrate nested structuring of java classes and their name conflict resolving issues
CO3	Implement inheritance, run time polymorphism and abstraction features of java in combination with java modular programming
CO4	Solve different run time and user inducted exception cases in the java program
CO5	Organize parallel processing scenarios with java multithread programming and incorporate them in web applications through java applet programming