COURSE STRUCTURE
AND
DETAILED SYLLABUS

COMPUTER SCIENCE AND ENGINEERING

For

B.TECH FOUR YEARS DEGREE PROGRAMME

(Applicable for the batches admitted from 2022-2023)

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
Bollikunta,Warangal -506005 Telangana State, India.

VAAGDEVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

COMPUTER SCIENCE AND ENGINEERING

Applicable from AY2022-23Batch

I Year I Semester

S.No.	Course Code	Course	L	T	P	Credits
1	B22MA01	Matrices and Calculus	3	1	0	4
2	B22CH01	Engineering Chemistry	3	1	0	4
3	B22CS01	Programming for Problem Solving	3	0	0	3
4	B22EE03	Basic Electrical Engineering	2	0	0	2
5	B22ME03	Computer Aided Engineering Graphics	1	0	4	3
6	B22CS02	Elements of Computer Science & Engineering	0	0	2	1
7	B22CH02	Engineering Chemistry Laboratory	0	0	2	1
8	B22CS03	Programming for Problem Solving Laboratory	0	0	2	1
9	B22EE04	Basic Electrical Engineering Laboratory	0	0	2	1
		Induction Program				
		Total	**12**	**2**	**12**	**20**

I Year II Semester

S.No.	Course Code	Course	L	T	P	Credits
1	B22MA02	Ordinary Differential Equations and Vector Calculus	3	1	0	4
2	B22PH01	Applied Physics	3	1	0	4
3	B22ME01	Engineering Workshop	0	1	3	2.5
4	B22EN01	English for Skill Enhancement	2	0	0	2
5	B22EC02	Electronic Devices and Circuits	2	0	0	2
6	B22CS04	Python Programming Laboratory	0	1	2	2
7	B22PH02	Applied Physics Laboratory	0	0	3	1.5
8	B22EN02	English Language and Communication Skills Laboratory	0	0	2	1
9	B22CS05	IT Workshop	0	0	2	1
10	B22CH03	Environmental Science	3	0	0	0
		Total	**13**	**4**	**12**	**20**

II YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	B22EC12	Digital Electronics	3	0	0	3
2	B22CS06	Data Structures	3	0	0	3
3		Computer Oriented Statistical Methods	3	1	0	4
4	B22CS07	Computer Organization and Architecture	3	0	0	3
5	B22CS08	Object Oriented Programming through Java	3	0	0	3
6	B22CS09	Data Structures Lab	0	0	3	1.5
7	B22CS10	Object Oriented Programming through Java Lab	0	0	3	1.5
8	B22DS01	Data visualization- R Programming/ Power BI	0	0	2	1
9		Gender Sensitization Lab	0	0	2	0
		Total	**15**	**1**	**10**	**20**

II YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	B22CS11	Discrete Mathematics	3	0	0	3
2	B22MB01	Business Economics & Financial Analysis	3	0	0	3
3	B22CS12	Operating Systems	3	0	0	3
4	B22CS13	Database Management Systems	3	0	0	3
5	B22CS14	Software Engineering	3	0	0	3
6	B22CS15	Operating Systems Lab	0	0	2	1
7	B22CS16	Database Management Systems Lab	0	0	2	1
8	B22CS17	Real-time Research Project/ Societal Related Project	0	0	4	2
9	B22CS18	Node JS/ React JS/ Django	0	0	2	1
10	*MC	Logical Reasoning and Quantitative Aptitude	3	0	0	0
		Total	**18**	**0**	**10**	**20**

III YEAR I SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1	B22CS27	Design and Analysis of Algorithms	3	1	0	4
2	B22CS28	Computer Networks	3	0	0	3
3	B22CS29	DevOps	3	0	0	3
4	**Professional Elective-I**					
	B22AI24	Quantum Computing	3	0	0	3
	B22CS30	Advanced Computer Architecture				
	B22AI11	Data Analytics				
	B22CS31	Image Processing				
	B22CS32	Principles of Programming Languages				
5	**Professional Elective -II**					
	B22CS33	Computer Graphics	3	0	0	3
	B22CS34	Embedded Systems				
	B22CS35	Information Retrieval Systems				
	B22CS36	Distributed Databases				
	B22AI12	Natural Language Processing				
6	B22CS37	Computer Networks Lab	0	0	2	1
7	B22CS38	DevOps Lab	0	0	2	1
8	B22EN03	Advanced Communication Skills Lab	0	0	2	1
9	B22CS39	UI design- Flutter	0	0	2	1
10	B22MB06	Intellectual Property Rights	3	0	0	0
		Total	**18**	**1**	**8**	**20**

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	B22AI05	Machine Learning	3	0	0	3
2	B22CS40	Formal Languages and Automata Theory	3	0	0	3
3	B22CS41	Artificial Intelligence	3	0	0	3
		Professional Elective – III				
4	B22CS42	Full Stack Development	3	0	0	3
	B22CS43	Internet of Things				
	B22CS44	Scripting Languages				
	B22CS45	Mobile Application Development				
	B22CS46	Software Testing Methodologies				
5		**Open Elective-I**	3	0	0	3
6	B22AI08	Machine Learning Lab	0	0	2	1
7	B22CS47	Artificial Intelligence Laboratory	0	0	2	1
		Professional Elective-III Lab				
8	B22CS48	Full Stack Development Lab	0	0	2	1
	B22CS49	Internet of Things Lab				
	B22CS50	Scripting Languages Lab				
	B22CS51	Mobile Application Development Lab				
	B22CS52	Software Testing Methodologies Lab				
9	B22CS53	Industrial Oriented Mini Project/ Internship/ Skill Development Course (Big data-Spark)	0	0	4	2
10	B22CH03	Environmental Science	3	0	0	0
		Total	**18**	**0**	**10**	**20**

Environmental Science in III Yr II Sem should be registered by Lateral Entry Students Only.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING

IV YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	B22CS54	Cryptography and Network Security	3	0	0	3
2	B22CS55	Compiler Design	3	0	0	3
3		**Professional Elective -IV**				
	B22CS56	Graph Theory	3	0	0	3
	B22CS57	Cyber Security				
	B22CS58	Soft Computing				
	B22AI23	Cloud Computing				
	B22CS59	Ad hoc & Sensor Networks				
		Professional Elective -V				
4	B22CS60	Advanced Algorithms	3	0	0	3
	B22CS61	Agile Methodology				
	B22CS62	Robotic Process Automation				
	B22CS63	Block chain Technology				
	B22CS64	Software Process & Project Management				
5		**Open Elective - II**	3	0	0	3
6	B22CS65	Cryptography and Network Security Lab	0	0	2	1
7	B22CS66	Compiler Design Lab	0	0	2	1
8	B22CS67	Project Stage - I	0	0	6	3
		Total Credits	**15**	**0**	**10**	**20**

IV YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	B22MB09	Organizational Behavior	3	0	0	3
		Professional Elective – VI				
2	B22CS68	Computational Complexity	3	0	0	3
	B22CS69	Distributed Systems				
	B22AI20	Deep Learning				
	B22CS70	Human Computer Interaction				
	B22CS71	Cyber Forensics				
3		**Open Elective – III**	3	0	0	3
4	B22CS75	Project Stage – II including Seminar	0	0	22	11
		Total Credits	**9**	**0**	**22**	**20**

***MC – Satisfactory/Unsatisfactory**
#Skill Course - 1 credit with 2 Practical Hours

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING

Open Electives:

S. No.	Course Code	Course Title	L	T	P	Credits
Open Elective – I						
1	B22CS11	Data Structures	3	0	0	3
2	B22CS13	Object Oriented Programming through Java				
3	B22CS18	Database Management Systems				
Open Elective – II						
1	B22CS17	Operating Systems	3	0	0	3
2	B22CS43	Internet of Things				
3	B22CS19	Software Engineering				
Open Elective – III						
1	B22CS72	Algorithms Design and Analysis	3	0	0	3
2	B22CS73	Introduction to Computer Networks				
3	B22CS74	Introduction to Cloud Computing				

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
MATRICES AND CALCULUS

B.Tech I Year I Sem. **L T P C**
3 1 0 4

Pre-requisites: Mathematical Knowledge at pre-university level

Course Objectives: To learn
- Types of matrices and their properties.
- Concept of a rank of the matrix and applying this concept to know the consistency and solving the system of linear equations.
- Concept of eigen values and eigenvectors and to reduce the quadratic form to canonical form
- Geometrical approach to the mean value theorems and their application to the mathematical problems
- Evaluation of surface areas and volumes of revolutions of curves.
- Evaluation of improper integrals using Beta and Gamma functions.
- Partial differentiation, concept of total derivative
- Finding maxima and minima of function of two and three variables.
- Evaluation of multiple integrals and their application

UNIT - I: Matrices **10 L**

Rank of a matrix by Echelon form and Normal form, Inverse of Non-singular matrices by Gauss-Jordan method, System of linear equations: Solving system of Homogeneous and Non-Homogeneous equations by Gauss elimination method, Gauss Seidel Iteration Method.

UNIT - II: Eigen values and Eigen vectors **10 L**

Linear Transformation and Orthogonal Transformation: Eigen values, Eigenvectors and their properties, Eigen values and Vectors with reference to Symmetric, Skew-symmetric, Hermitian, Skew- Hermitian, orthogonal and Unitary Matrices. Diagonalization of a matrix, Cayley-Hamilton Theorem (without proof), finding inverse and power of a matrix by Cayley-Hamilton Theorem,
Quadratic Forms: Quadratic forms and Nature of the Quadratic Forms, Reduction of Quadratic form to canonical forms.

UNIT - III: Calculus **10 L**

Mean value theorems: Rolle's theorem, Lagrange's Mean value theorem with their Geometrical Interpretation and applications, Cauchy's Mean value Theorem, Taylor's Series. Applications of definite integrals to evaluate surface areas and volumes of revolutions of curves (Only in Cartesian coordinates),
Beta and Gamma Functions: Introduction to Improper Integrals, Definition of Beta and Gamma functions, properties and other forms. Relation between Beta and Gamma functions. Evaluation of Improper integrals using Beta and Gamma functions

UNIT - IV: Multivariable Calculus (Partial Differentiation and applications) 10 L

Definitions of Limit and continuity. Partial Differentiation: Euler's Theorem, Total derivative, Jacobian, Functional dependence & independence. Applications: Maxima and minima of functions of two variables and three variables using method of Lagrange multipliers.

UNIT-V: Multivariable Calculus (Integration) **8 L**

Evaluation of Double Integrals (Cartesian and polar coordinates), change of order of integration (only

Cartesian form), Evaluation of Triple Integrals: Change of variables (Cartesian to polar) for double and (Cartesian to Spherical and Cylindrical polar coordinates) for triple integrals.

Course outcomes: After learning the contents of this paper the student must be able to

CO-1: Write the matrix representation of a set of linear equations and to analyse the solution of the system of equations

CO-2: Find the Eigen values and Eigen vectors. Reduce the quadratic form to canonical form using orthogonal transformations.

CO-3: Solve the applications on the mean value theorems. and Evaluate the improper integrals using Beta and Gamma functions.

CO-4: Find the extreme values of functions of two variables with/ without constraints

CO-5: Evaluate the multiple integrals and apply the concept to find areas, volumes

TEXT BOOKS:

1. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.
2. R.K. Jain and S.R.K. Iyengar, Advanced Engineering Mathematics, Narosa Publications, 5th Editon, 2016.

REFERENCE BOOKS:

1. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9thEdition,Pearson, Reprint, 2002.
3. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
4. H. K. Dass and Er. RajnishVerma, Higher Engineering Mathematics, S Chand and Company Limited, New Delhi.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
ENGINEERING CHEMISTRY

B.Tech I Year I Sem. **L T P C**
3 1 0 4

Pre-requisites: Chemistry Knowledge at pre-university level

Course Objectives:

- To bring adaptability to new developments in Engineering Chemistry and to acquire the skills required to become a perfect engineer.
- To include the importance of water in industrial usage, fundamental aspects of battery chemistry, significance of corrosion it's control to protect the structures.
- To imbibe the basic concepts of petroleum and its products.
- To acquire required knowledge about engineering materials like cement, smart materials and Lubricants.

UNIT - I: Water and its treatment: [8]

Introduction to hardness of water – Estimation of hardness of water by complexometric method and related numerical problems. Potable water and its specifications - Steps involved in the treatment of potable water - Disinfection of potable water by chlorination and break - point chlorination. Defluoridation of water by Nalgonda technique.

Boiler troubles: Sludges, Scales and Caustic embrittlement. Internal treatment of Boiler feed water - Calgon conditioning - Phosphate conditioning - Colloidal conditioning, External treatment methods - Softening of water by ion- exchange processes. Desalination of water – Reverse osmosis.

UNIT – II Battery Chemistry & Corrosion [**8**]

Introduction - Classification of batteries- primary, secondary and reserve batteries with examples. Characteristics of batteries. Construction, working and applications of: Zn-air and Lithium ion battery, Applications of Li-ion battery to electrical vehicles. Fuel Cells- Differences between battery and a fuel cell, Construction and applications of Methanol Oxygen fuel cell and Solid oxide fuel cell. Solar cells - Introduction and applications of Solar cells.

Corrosion: Causes and effects of corrosion – theories of chemical and electrochemical corrosion – mechanism of electrochemical corrosion, Types of corrosion: Galvanic, water-line and pitting corrosion. Factors affecting rate of corrosion, Corrosion control methods- Cathodic protection – Sacrificial anode and impressed current methods.

UNIT - III: Polymeric materials: [8]

Definition – Classification of polymers with examples – Types of polymerization – addition (free radical addition) and condensation polymerization with examples – Nylon 6:6, Terylene.

Plastics: Definition and characteristics- thermoplastic and thermosetting plastics, Preparation, Properties and engineering applications of PVC and Bakelite, Teflon.

Rubbers: Natural rubber and its vulcanization.

Elastomers: Characteristics –preparation – properties and applications of Buna-S, Butyl and Thiokol rubber.

Conducting polymers: Characteristics and Classification with examples-mechanism of conduction in trans-polyacetylene and applications of conducting polymers.

Biodegradable polymers: Concept and advantages - Polylactic acid and poly vinyl alcohol and their applications.

UNIT - IV: Energy Sources: [8]

Introduction, Calorific value of fuel – HCV, LCV- Dulongs formula. Classification- solid fuels: coal – analysis of coal – proximate and ultimate analysis and their significance. Liquid fuels – petroleum and its refining, cracking types – moving bed catalytic cracking. Knocking – octane and cetane rating, synthetic petrol - Fischer-Tropsch's process; Gaseous fuels – composition and uses of natural gas, LPG and CNG, Biodiesel – Transesterification, advantages.

UNIT - V: **Engineering Materials: [8]**

Cement: Portland cement, its composition, setting and hardening.

Smart materials and their engineering applications

Shape memory materials- Poly L- Lactic acid. Thermo response materials- Polyacryl amides, Poly vinyl amides

Lubricants: Classification of lubricants with examples-characteristics of a good lubricants - mechanism of lubrication (thick film, thin film and extreme pressure)- properties of lubricants: viscosity, cloud point, pour point, flash point and fire point.

Course Outcomes:

CO-1: The students are able to understand the basic properties of water and its usage in domestic and industrial purposes.

CO-2: Students will acquire the basic knowledge of electrochemical procedures related to corrosion and its control.

CO-3: They can learn the fundamentals and general properties of polymers and other engineering materials.

CO-4: They can predict potential applications of chemistry and practical utility in order to become good engineers and entrepreneurs.

CO-5: Appreciate the features and applications of lubricants and smart materials.

TEXT BOOKS:

1. Engineering Chemistry by P.C. Jain and M. Jain, Dhanpatrai Publishing Company, 2010
2. Engineering Chemistry by Rama Devi, Venkata Ramana Reddy and Rath, Cengage learning, 2016
3. A text book of Engineering Chemistry by M. Thirumala Chary, E. Laxminarayana and K. Shashikala, Pearson Publications, 2021.
4. Textbook of Engineering Chemistry by Jaya Shree Anireddy, Wiley Publications.

REFERENCE BOOKS:

1. Engineering Chemistry by Shikha Agarwal, Cambridge University Press, Delhi (2015)
2. Engineering Chemistry by Shashi Chawla, Dhanpatrai and Company (P) Ltd. Delhi

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
PROGRAMMING FOR PROBLEM SOLVING

B.Tech I Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

- To learn the fundamentals of computers.
- To understand the various steps in program development.
- To learn the syntax and semantics of the C programming language.
- To learn the usage of structured programming approaches in solving problems.

UNIT - I: Introduction to Programming

Compilers, compiling and executing a program.

Representation of Algorithm - Algorithms for finding roots of a quadratic equations, finding minimum and maximum numbers of a given set, finding if a number is prime number Flowchart/Pseudocode with examples, Program design and structured programming

Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code, Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators

Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto, Iteration with for, while, do-while loops

I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments

UNIT - II: Arrays, Strings, Structures and Pointers:

Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings

Structures: Defining structures, initializing structures, unions, Array of structures

Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of Pointers in self- referential structures, usage of self referential structures in linked list (no implementation) Enumeration data type

UNIT - III: Preprocessor and File handling in C:

Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef Files: Text and Binary files, Creating and Reading and writing text and binary files, Appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

UNIT - IV: Function and Dynamic Memory Allocation:

Functions: Designing structured programs, Declaring a function, Signature of a function,

Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries

Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

UNIT - V: Searching and Sorting:

Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms), Basic concept of order of complexity through the example programs

Course Outcomes: The student will learn

CO-1: To write algorithms and to draw flowcharts for solving problems. To convert the algorithms/flowcharts to C programs.
CO-2: To use arrays, pointers, strings and structures to write C programs.
CO-3: Ability to design and implement different types of file structures using standard methodology.
CO-4: To decompose a problem into functions and to develop modular reusable code.
CO-5: Searching and sorting problems.

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
BASIC ELECTRICAL ENGINEERING

B.Tech I Year I Sem.

L	T	P	C
2	0	0	2

Prerequisites: Mathematics

Course Objectives:

- To understand DC and Single & Three phase AC circuits
- To study and understand the different types of DC, AC machines and Transformers.
- To import the knowledge of various electrical installations and the concept of power, power factor and its improvement.

UNIT-I:

D.C. Circuits: Electrical circuit elements (R, L and C), voltage and current sources, KVL&KCL, analysis of simple circuits with dc excitation. Superposition, Thevenin and Norton Theorems. Time-domain analysis of first-order RL and RC circuits.

UNIT-II:

A.C. Circuits: Representation of sinusoidal waveforms, peak and rms values, phasor representation, real power, reactive power, apparent power, power factor, Analysis of single-phase ac circuits consisting of R, L, C, RL, RC, RLC combinations (series and parallel), resonance in series R-L-C circuit. Three-phase balanced circuits, voltage and current relations in star and delta connections.

UNIT-III:

Transformers: Ideal and practical transformer, equivalent circuit, losses in transformers, regulation and efficiency. Auto-transformer and three-phase transformer connections.

UNIT-IV:

Electrical Machines: Construction and working principle of dc machine, performance characteristics of dc shunt machine. Generation of rotating magnetic field, Construction and working of a three-phase induction motor, Significance of torque-slip characteristics. Single-phase induction motor, Construction and working. Construction and working of synchronous generator.

UNIT-V:

Electrical Installations: Components of LT Switchgear: Switch Fuse Unit (SFU), MCB, ELCB, MCCB, Types of Wires and Cables, Earthing. Types of Batteries, Important Characteristics for Batteries. Elementary calculations for energy consumption, power factor improvement and battery backup.

Course Outcomes: After learning the contents of this paper the student must be able to

1. Analyze circuit theorems, mesh and nodal analysis, series and parallel networks, Electrical power
2. Gain knowledge on AC circuits, reactance, Impedance, Susceptance and Admittance and Power Factor
3. Learn the working principle of DC motors, Transformers
4. Understand the construction and performance characteristics of Electrical Machines
5. Introduce components of Low Voltage Electrical Installations

TEXT BOOKS:

1. D.P. Kothari and I. J. Nagrath, "Basic Electrical Engineering", Tata McGraw Hill, 4th Edition, 2019.
2. MS Naidu and S Kamakshaiah, "Basic Electrical Engineering", Tata McGraw Hill, 2nd Edition, 2008.

REFERENCE BOOKS:

1. P. Ramana, M. Suryakalavathi, G.T. Chandrasheker, "Basic Electrical Engineering", S. Chand, 2nd Edition, 2019.
2. D. C. Kulshreshtha, "Basic Electrical Engineering", McGraw Hill, 2009
3. M. S. Sukhija, T. K. Nagsarkar, "Basic Electrical and Electronics Engineering", Oxford, 1st Edition, 2012.
4. Abhijit Chakrabarthi, Sudipta Debnath, Chandan Kumar Chanda, "Basic Electrical Engineering", 2nd Edition, McGraw Hill, 2021.
5. L. S. Bobrow, "Fundamentals of Electrical Engineering", Oxford University Press, 2011.
6. E. Hughes, "Electrical and Electronics Technology", Pearson, 2010.
7. V. D. Toro, "Electrical Engineering Fundamentals", Prentice Hall India, 1989.

ONLINE RESOURCES

1. Basic Electrical Circuits - Course (nptel.ac.in)
2. https://nptel.ac.in/courses/108105155

******

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
COMPUTER AIDED ENGINEERING GRAPHICS

B.Tech I Year I Sem. **L T P C**
1 0 4 3

COURSE OBJECTIVES:

The objectives of this course are to

- To develop the ability of visualization of different objects through technical drawings.
- To acquire computer drafting skill for communication of concepts, ideas in the design of engineering products.

UNIT - I:

Introduction to Engineering Graphics: Principles of Engineering Graphics and their Significance, Scales – Plain & Diagonal, Conic Sections including the Rectangular Hyperbola – General method only. Cycloid, Epicycloid and Hypocycloid, Introduction to Computer aided drafting – views, commands and conics

UNIT- II:

Orthographic Projections: Principles of Orthographic Projections – Conventions – Projections of Points and Lines, Projections of Plane regular geometric figures. Auxiliary Planes. Computer aided orthographic projections – points, lines and planes

UNIT – III:

Projections of Regular Solids – Auxiliary Views - Sections or Sectional views of Right Regular Solids –Prism, Cylinder, Pyramid, Cone – Auxiliary views, Computer aided projections of solids – sectional views

UNIT – IV:

Development of Surfaces of Right Regular Solids – Prism, Cylinder, Pyramid and Cone, Development of surfaces using computer aided drafting

UNIT – V:

Isometric Projections: Principles of Isometric Projection – Isometric Scale – Isometric Views – Conventions – Isometric Views of Lines, Plane Figures, Simple and Compound Solids – Isometric Projection of objects having non- isometric lines. Isometric Projection of Spherical Parts. Conversion of Isometric Views to Orthographic Views and Vice-versa –Conventions. Conversion of orthographic projection into isometric view using computer aided drafting.

Course Outcomes: At the end of the course, the student will be able to:

CO 1: Apply computer aided drafting tools to create 2D and 3D objects sketch conics and different types of solids

CO 2: Appreciate the need of Sectional views of solids and Development of surfaces of solids

CO 3: Read and interpret engineering drawings

CO 4: Conversion of orthographic projection into isometric view and vice versa manually and by using computer aided drafting

TEXT BOOKS:

1. Engineering Drawing N.D. Bhatt / Charotar
2. Engineering Drawing and graphics Using AutoCAD Third Edition, T. Jeyapoovan, Vikas: S.Chand and company Ltd.

REFERENCE BOOKS:

1. Engineering Drawing, Basant Agrawal and C M Agrawal, Third Edition McGraw Hill
2. Engineering Graphics and Design, WILEY, Edition 2020
3. Engineering Drawing, M. B. Shah, B.C. Rane / Pearson.
4. Engineering Drawing, N. S. Parthasarathy and Vela Murali, Oxford
5. Computer Aided Engineering Drawing – K Balaveera Reddy et al – CBS Publishers

NOTE: - External examination is conducted in conventional mode and internal evaluation to be done by both conventional as well as using computer aided drafting.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
ELEMENTS OF COMPUTER SCIENCE AND ENGINEERING

B.Tech I Year I Sem. **L T P C**
0 0 2 1

Course Objective: To provide an overview of the subjects of computer science and engineering.

UNIT – I

Basics of a Computer – Hardware, Software, Generations of computers. Hardware - functional units, Components of CPU, Memory – hierarchy, types of memory, Input and output devices. Software – systems software, application software, packages, frameworks, IDEs.

UNIT – II

Software development – waterfall model, Agile, Types of computer languages – Programming, markup, scripting Program Development – steps in program development, flowcharts, algorithms, data structures – definition, types of data structures

UNIT – III

Operating systems: Functions of operating systems, types of operating systems, Device & Resource management

Database Management Systems: Data models, RDBMS, SQL, Database Transactions, data centers, cloud services

UNIT – IV

Computer Networks: Advantages of computer networks, LAN, WAN, MAN, internet, WiFi, sensor networks, vehicular networks, 5G communication.

World Wide Web – Basics, role of HTML, CSS, XML, Tools for web designing, Social media, Online social networks.

Security – information security, cyber security, cyber laws

UNIT – V

Autonomous Systems: IoT, Robotics, Drones, Artificial Intelligence – Learning, Game Development, natural language processing, image and video processing. Cloud Basics

Course Outcomes:

CO-1: Know the working principles of functional units of a basic Computer
CO-2: Understand program development, the use of data structures and algorithms in problem solving.
CO-3: Know the need and types of operating system, database systems.
CO-4: Understand the significance of networks, internet, WWW and cyber security.
CO-5: Understand Autonomous systems, the application of artificial intelligence.

TEXT BOOK:

1. Invitation to Computer Science, G. Michael Schneider, Macalester College, Judith L. Gersting University of Hawaii, Hilo, Contributing author: Keith Miller University of Illinois, Springfield.

REFERENCE BOOKS:

1. Fundamentals of Computers, Reema Thareja, Oxford Higher Education, Oxford University Press.
2. Introduction to computers, Peter Norton, 8th Edition, Tata McGraw Hill.
3. Computer Fundamentals, Anita Goel, Pearson Education India, 2010.
4. Elements of computer science, Cengage.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
ENGINEERING CHEMISTRY LABORATORY

B.Tech I Year I Sem. **L T P C**
0 0 2 1

Course Objectives: The course consists of experiments related to the principles of chemistry required for engineering student. The student will learn:

- Estimation of hardness and chloride content of water to check its suitability for drinking purpose.
- Students are able to perform estimations of acids and bases using conductometry, potentiometry methods.
- Students will learn to prepare polymers such as Bakelite and nylon-6,6 in the laboratory.
- Students will learn skills related to the lubricant properties such as saponification value and viscosity of oils.

List of Experiments:

I. **Volumetric Analysis:** Estimation of Hardness of water by EDTA Complexometry method.

II. **Conductometry:** Estimation of the concentration of an acid by Conductometry.

III. **Potentiometry:** Estimation of the amount of Fe^{+2} by Potentiomentry.

IV. **Determination of P^H**: Determination of P^H of unknown acid solution by using Quinhydrone electrode.

V. **Preparations:**

1. Preparation of Bakelite.
2. Preparation Nylon - 6,6.

VI. **Lubricants:**

1. Estimation of acid value of given lubricant oil.
2. Estimation of Viscosity of lubricant oil using Ostwald's Viscometer.

VII. Determination of surface tension of a given liquid using Stalagmometer.

VIII. **Virtual lab experiments**

1. Construction of Fuel cell and it's working.
2. Smart materials for Biomedical applications
3. Batteries for electrical vehicles.
4. Functioning of solar cell and its applications.

Course Outcomes: The experiments will make the student gain skills on:

CO-1: Able to determine the hardness of water

CO-2: Able to perform methods such as conductometry, and potentiometry in order to find out the concentrations or equivalence points of acid, and P^H of unknown solutions.

CO-3: Students are able to prepare polymers like bakelite and nylon-6,6.

CO-4: Estimations saponification value, and viscosity of lubricant oils.

REFERENCE BOOKS:

1. Lab manual for Engineering chemistry by B. Ramadevi and P. Aparna, S Chand Publications, New Delhi (2022)
2. Vogel's text book of practical organic chemistry 5th edition
3. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications.
4. College Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd. New Delhi (2007).

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
PROGRAMMING FOR PROBLEM SOLVING LABORATORY

B.Tech I Year I Sem. **L T P C**
0 0 2 1

[Note:The programs may be executed using any available Open Source/ Freely available IDE Some of the Tools available are:
CodeLite: https://codelite.org/
Code:Blocks:
http://www.codeblocks.org/
DevCpp :
http://www.bloodshed.net/devcpp.html
Eclipse: http://www.eclipse.org
This list is not exhaustive and is NOT in any order of preference]

Course Objectives: The students will learn the following:

- To work with an IDE to create, edit, compile, run and debug programs
- To analyze the various steps in program development.
- To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
- To develop modular, reusable and readable C Programs using the concepts like functions, arrays etc.
- To Write programs using the Dynamic Memory Allocation concept.
- To create, read from and write to text and binary files

Practice sessions:

a. Write a simple program that prints the results of all the operators available in C (including pre/ post increment , bitwise and/or/not , etc.). Read required operand values from standard input.
b. Write a simple program that converts one given data type to another using auto conversion and casting. Take the values from standard input.

Simple numeric problems:

a. Write a program for finding the max and min from the three numbers.
b. Write the program for the simple, compound interest.
c. Write a program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input.
d. Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:
   5 x 1 = 5
   5 x 2 = 10
   5 x 3 = 15

e. Write a program that shows the binary equivalent of a given positive number between 0 to 255.

Expression Evaluation:

a. A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula s = ut+(1/2)at^2 where u and a are the initial velocity in m/sec (= 0) and acceleration in m/sec^2 (= 9.8 m/s^2)).
b. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,*, /, % and use Switch Statement)
c. Write a program that finds if a given number is a prime number
d. Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
e. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
f. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
g. Write a C program to find the roots of a Quadratic equation.
h. Write a C program to calculate the following, where x is a fractional value.

1-x/2+x^2/4-x^3/6

j. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: 1+x+x^2+x^3+…… +x^n. For example: if n is 3 and x is 5, then
the program computes 1+5+25+125.

Arrays, Pointers and Functions:

a. Write a C program to find the minimum, maximum and average in an array of integers.
b. Write a function to compute mean, variance, Standard Deviation, sorting of n elements in a single dimension array.
c. Write a program to count the occurrence of an element in an array.
d. Write a C program that uses functions to perform the following:
e. Addition of Two Matrices
f. Multiplication of Two Matrices
g. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be the same.
h. Write C programs that use both recursive and non-recursive functions
i. To find the factorial of a given integer.
j. To find the GCD (greatest common divisor) of two given integers.
k. To find x^n
l. Write C program to demonstrate call by value and call by reference.
m. Write a program to pass a single dimensional array as a parameter to a function.
n. Write a program for reading elements using a pointer into an array and display the values using the array.
o. Write a program for display values reverse order from an array using a pointer.

p. Write a program through a pointer variable to sum of n elements from an array.

Files:

a. Write a C program to display the contents of a file to standard output device.
b. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
c. Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.
d. Write a C program that does the following:
   It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function)
   Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function)
   The program should then read all 10 values and print them back.
e. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

Strings:

a. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
b. Write a C program that converts a number ranging from 1 to 50 to Roman equivalent
c. Write a C program that uses functions to perform the following operations:
d. To insert a sub-string into a given main string from a given position.
e. To delete n Characters from a given position in a given string.
f. Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)
g. Write a C program that displays the position of a character ch in the string S or – 1 if S doesn't contain ch.
h. Write a C program to count the lines, words and characters in a given text.
i. Write a program to read 'n' strings and search the given substring.
j. Write a program to sort the strings in logical order.

Miscellaneous:

a. Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

b. Write a C program to construct a pyramid of numbers as follows:

```
1          *          1          1          *
1 2        * *        2 3        2 2        * *
1 2 3      * * *      4 5 6      3 3 3      * *
                                             *
                                4 4 4 4    * *
                                           *
```

Sorting and Searching:

a. Write a C program that uses non recursive function to search for a Key value in a given
b. list of integers using linear search method.
c. Write a C program that uses non recursive function to search for a Key value in a given
d. sorted list of integers using binary search method.
e. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.
f. Write a C program that sorts the given array of integers using selection sort in descending order
g. Write a C program that sorts the given array of integers using insertion sort in ascending order
h. Write a C program that sorts a given array of names

Course Outcomes: The candidate is expected to be able to:

CO-1: Understand basic structure of the C Programming, data types, declaration and usage of variables, control structures and all related concepts.
CO-2: Ability to understand any algorithm and Write the C programming code in executable form.
CO-3: Implement Programs using functions, pointers and arrays, and use the pre-processors to solve real time problems.
CO-4: Ability to use file structures and implement programs on files and Implement programs on sorting and searching techniques.

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, PHI
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
BASIC ELECTRICAL ENGINEERING LABORATORY

B.Tech I Year I Sem. **L T P C**
0 0 2 1

Prerequisites: Basic Electrical Engineering

Course Objectives:

- To measure the electrical parameters for different types of DC and AC circuits using conventional and theorems approach.
- To study the transient response of various R, L and C circuits using different excitations.https://docs.google.com/document/d/1xSjf3zaXxvVVZYY4doUmPC6qOJ2egSKLOh_T0-ap3Wc/edit?usp=sharing
- To determine the performance of different types of DC, AC machines and Transformers.

List of experiments/demonstrations:

1. Verification of KVL and KCL
2. Verification of Thevenin's and Norton's theorem
3. Transient Response of Series RL and RC circuits for DC excitation
4. Resonance in series RLC circuit
5. Calculations and Verification of Impedance and Current of RL, RC and RLC series circuits
6. Measurement of Voltage, Current and Real Power in primary and Secondary Circuits of a Single-Phase Transformer
7. Performance Characteristics of a DC Shunt Motor
8. Torque-Speed Characteristics of a Three-phase Induction Motor.
9. Verification of Superposition theorem.
10. Load Test on Single Phase Transformer (Calculate Efficiency and Regulation)

Course Outcomes: After learning the contents of this paper the student must be able to

1. Verify the basic electrical circuits through different laws and theorems.

2. Analyze the transient responses of R, L and C circuits for DC excitation.

3. Create resonance condition in series R-L-C circuit.

4. Analyze the performance of DC shunt motor, single phase transformer and three Phase Induction Motor.

TEXT BOOKS:

1. D.P. Kothari and I. J. Nagrath, "Basic Electrical Engineering", Tata McGraw Hill, 4th Edition, 2019.
2. MS Naidu and S Kamakshaiah, "Basic Electrical Engineering", Tata McGraw Hill, 2nd Edition, 2008.

REFERENCE BOOKS:

1. P. Ramana, M. Suryakalavathi, G.T.Chandrasheker,"Basic Electrical Engineering", S. Chand, 2nd Edition, 2019.
2. D. C. Kulshreshtha, "Basic Electrical Engineering", McGraw Hill, 2009
3. M. S. Sukhija, T. K. Nagsarkar, "Basic Electrical and Electronics Engineering", Oxford, 1st Edition, 2012.
4. Abhijit Chakrabarthi, Sudipta Debnath, Chandan Kumar Chanda, "Basic Electrical Engineering", 2nd Edition, McGraw Hill, 2021.
5. L. S. Bobrow, "Fundamentals of Electrical Engineering", Oxford University Press, 2011.
6. E. Hughes, "Electrical and Electronics Technology", Pearson, 2010.
7. V. D. Toro, "Electrical Engineering Fundamentals", Prentice Hall India, 1989.

******

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
ORDINARY DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS

B.Tech I Year II Sem. **L T P C**
3 1 0 4

Pre-requisites: Mathematical Knowledge at pre-university level

Course Objectives: To learn
- Methods of solving the differential equations of first and higher order.
- Concept, properties of Laplace transforms
- Solving ordinary differential equations using Laplace transforms techniques.
- The physical quantities involved in engineering field related to vector valued functions
- The basic properties of vector valued functions and their applications to line, surface and volume integrals.

UNIT-I: First Order ODE **8 L**

Exact differential equations, Equations reducible to exact differential equations, linear and Bernoulli's equations, Orthogonal Trajectories (only in Cartesian Coordinates). Applications: Newton's law of cooling, Law of natural growth and decay.

UNIT-II: Ordinary Differential Equations of Higher Order **10 L**

Second order linear differential equations with constant coefficients: Non-Homogeneous terms of the type $e^{ax}, \sin ax, \cos ax,$ polynomials in $x, e^{axy}V(x)$ and $xV(x)$ method of variation of parameters, Equations reducible to linear ODE with constant coefficients: Legendre's equation, Cauchy-Euler equation. Applications: Electrical Circuits (Both first and second order).

UNIT-III: Laplace transforms **10 L**

Laplace Transforms: Laplace Transform of standard functions, First shifting theorem, Second shifting theorem, Unit step function, Dirac delta function, Laplace transforms of functions when they are multiplied and divided by 't', Laplace transforms of derivatives and integrals of function, Evaluation of integrals by Laplace transforms, Laplace transform of periodic functions, Inverse Laplace transform by different methods, convolution theorem (without proof). Applications: solving differential equations with constant coefficients with give conditions by Laplace Transform method.

UNIT-IV: Vector Differentiation **10 L**

Vector point functions and scalar point functions, Gradient, Divergence and Curl, Directional derivatives, , Vector Identities(without proofs) Tangent plane and normal line, Scalar potential functions, Solenoidal and Irrotational vectors.

UNIT-V: Vector Integration **10 L**

Line, Surface and Volume Integrals, Theorems of Green, Gauss and Stokes (without proofs) and their applications.

Course outcomes: After learning the contents of this paper the student must be able to
1. Identify whether the given differential equation of first order is exact or not
2. Solve higher differential equation and apply the concept of differential equation to real world problems.
3. Extend the basic concepts of differential calculus to vector functions in a simple and natural fashion.
4. Extend the basic concepts of differential calculus to vector functions in a simple and natural fashion.
5. Evaluate the line, surface and volume integrals and converting them from one to another.

TEXT BOOKS:

1. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010

2. R.K. Jain and S.R.K. Iyengar, Advanced Engineering Mathematics, Narosa Publications, 5th Edition, 2016.

REFERENCE BOOKS:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
3. H. K. Dass and Er. RajnishVerma, Higher Engineering Mathematics, S Chand and Company Limited, New Delhi.
4. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
APPLIED PHYSICS

B.Tech I Year II Sem. **L T P C**
3 1 0 4

Pre-requisites: 10 + 2 Physics

Course Objectives: The objectives of this course for the student are to:
- Understand the basic principles of quantum physics and band theory of solids.
- Understand the underlying mechanism involved in construction and working principles of various semiconductor devices.
- Study the fundamental concepts related to the dielectric, magnetic and energy materials.
- Identify the importance of nanoscale, quantum confinement and various fabrications techniques.
- Study the characteristics of lasers and optical fibres.

UNIT - I: QUANTUM PHYSICS AND SOLIDS
QUANTUM MECHANICS: Introduction To Quantum Physics - Blackbody Radiation – Stefan-Boltzmann's Law, Wein's And Rayleigh-Jean's Law, Planck's Radiation Law (qualitative) - Photoelectric Effect- waves and particles – de Broglie hypothesis – properties of matter waves- Davisson And Germer Experiment –Heisenberg Uncertainty Principle - Born Interpretation Of The Wave Function – Time Independent Schrodinger Wave Equation - Particle in One Dimensional Potential Box – **SOLIDS:** Free Electron Theory (Drude & Lorentz, Sommerfeld) - Fermi-Dirac Distribution - Bloch's Theorem -Kronig-Penney Model (qualitative) - E-K Diagram- Effective Mass Of electron- Origin Of Energy Bands- Classification Of Solids.

UNIT - II: SEMICONDUCTORS AND DEVICES
Intrinsic And Extrinsic Semiconductors – Hall Effect - Direct And Indirect Band Gap Semiconductors - Construction, Principle , Operation And Characteristics Of P-N Junction Diode, Zener Diode And Bipolar Junction Transistor (BJT)–LED, PIN Diode, Avalanche Photo Diode (APD) And Solar Cells, Their Structure, Materials, Working Principle And Characteristics.

UNIT - III: DIELECTRIC, MAGNETIC AND ENERGY MATERIALS
DIELECTRIC MATERIALS: Basic Definitions- Types of Polarizations (Qualitative) - Ferroelectric, Piezoelectric, and Pyroelectric Materials – Applications – Liquid Crystal Displays (LCD) And Crystal Oscillators. **MAGNETIC MATERIALS:** Hysteresis - Soft And Hard Magnetic Materials - Magnetostriction, Magnetoresistance - Applications - Bubble Memory Devices, Magnetic Field Sensors And Multi-Ferroics. **ENERGY MATERIALS**: Conductivity of Liquid and Solid Electrolytes-Superionic Conductors - Materials And electrolytes for Super Capacitors - Rechargeable Ion Batteries, Solid Fuel Cells.

UNIT - IV: NANOTECHNOLOGY
Nanoscale, Quantum Confinement, Surface to Volume Ratio, Bottom-Up Fabrication: Sol-Gel, Precipitation, Combustion Methods – Top-Down Fabrication: Ball Milling - Physical Vapor Deposition (PVD) - Chemical Vapor Deposition (CVD) - Characterization Techniques - XRD, SEM &TEM - Applications of Nano materials.

UNIT - V: LASER AND FIBER OPTICS
LASERS: Laser Beam Characteristics-Three Quantum Processes-Einstein Coefficients And Their Relations- Lasing Action - Pumping Methods- Ruby Laser, He-Ne Laser, Nd: YAG Laser- Semiconductor Laser-Applications Of Laser. **FIBER OPTICS**: Introduction To Optical Fiber- Advantages Of Optical Fibers - Total Internal Reflection construction of Optical Fiber - Acceptance Angle - Numerical Aperture-Classification Of Optical Fibers losses in Optical Fiber - Optical Fiber For Communication System - Applications.

Course Outcomes: At the end of the course the student will be able to:

CO-1: Understand physical world from fundamental point of view by the concepts of Quantum Mechanics and visualize the difference between conductor, semiconductor,and an insulator by classification of solids

CO-2: Identify the role of semiconductor devices in science and engineering Applications.

CO-3: Explore the fundamental properties of dielectric, magnetic materials and energy for their applications.

CO-4: Appreciate the features and applications of Nano materials.

CO-5: Understand various aspects of Lasers and Optical fibre and their applications in diverse fields.

TEXT BOOKS:

1. M. N. Avadhanulu, P.G. Kshirsagar& TVS Arun Murthy" A Text book of Engineering Physics"-S. Chand Publications, 11th Edition 2019.
2. Engineering Physics by Shatendra Sharma and Jyotsna Sharma, Pearson Publication,2019
3. Semiconductor Physics and Devices- Basic Principle – Donald A, Neamen, Mc Graw Hill,4thEdition,2021.
4. B.K. Pandey and S. Chaturvedi, Engineering Physics, Cengage Learning, 2ndEdition,2022.
5. Essentials of Nanoscience & Nanotechnology by Narasimha Reddy Katta, Typical Creatives NANO DIGEST, 1st Edition, 2021.

REFERENCE BOOKS:

1. Quantum Physics, H.C. Verma, TBS Publication, 2nd Edition 2012.
2. Fundamentals of Physics – Halliday, Resnick and Walker, John Wiley &Sons, 11th Edition,2018.
3. Introduction to Solid State Physics, Charles Kittel, Wiley Eastern, 2019.
4. Elementary Solid State Physics, S.L. Gupta and V. Kumar, PragathiPrakashan, 2019.
5. A.K. Bhandhopadhya - Nano Materials, New Age International, 1stEdition, 2007.
6. Energy Materials a Short Introduction to Functional Materials for Energy Conversion and Storage Aliaksandr S. Bandarenka, CRC Press Taylor & Francis Group
7. Energy Materials, Taylor & Francis Group, 1st Edition

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
ENGINEERING WORKSHOP

B.Tech I Year II Sem. **L T P C**
0 1 3 2.5

Pre-requisites: Practical skill

Course Objectives:
- To Study of different hand operated power tools, uses and their demonstration.
- To gain a good basic working knowledge required for the production of various engineering products.
- To provide hands on experience about use of different engineering materials, tools, equipments and processes those are common in the engineering field.
- To develop a right attitude, team working, precision and safety at work place.
- It explains the construction, function, use and application of different working tools, equipment and machines.
- To study commonly used carpentry joints.
- To have practical exposure to various welding and joining processes.
- Identify and use marking out tools, hand tools, measuring equipment and to work to prescribed tolerances.

1. TRADES FOR EXERCISES:
At least two exercises from each trade:

I. Carpentry – (T-Lap Joint, Dovetail Joint, Mortise & Tenon Joint)
II. Fitting – (V-Fit, Dovetail Fit & Semi-circular fit)
III. Tin-Smithy – (Square Tin, Rectangular Tray & Conical Funnel)
IV. Foundry – (Preparation of Green Sand Mould using Single Piece and Split Pattern)
V. Welding Practice – (Arc Welding & Gas Welding)
VI. House-wiring – (Parallel & Series, Two-way Switch and Tube Light)
VII. Black Smithy – (Round to Square, Fan Hook and S-Hook)

2. TRADES FOR DEMONSTRATION & EXPOSURE:
Plumbing, Machine Shop, Metal Cutting (Water Plasma), Power tools in construction and Wood Working.

Course Outcomes: At the end of the course, the student will be able to:
CO 1: Study and practice on machine tools and their operations
CO 2: Practice on manufacturing of components using workshop trades including pluming, fitting, carpentry, foundry, house wiring and welding.
CO 3: Identify and apply suitable tools for different trades of Engineering processes including drilling, material removing, measuring, chiseling.
CO 4: Apply basic electrical engineering knowledge for house wiring practice.

TEXT BOOKS:
1. Workshop Practice /B. L. Juneja / Cengage
2. Workshop Manual / K. Venugopal / Anuradha.

REFERENCE BOOKS:
1. Work shop Manual - P. Kannaiah/ K.L. Narayana/ Scitech
2. Workshop Manual / Venkat Reddy/ BSP

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
ENGLISH FOR SKILL ENHANCEMENT

B.Tech I Year II Sem. **L T P C**
2 0 0 2

Course Objectives: This course will enable the students to:

- Improve the language proficiency of students in English with an emphasis on Vocabulary, Grammar, Reading and Writing skills.
- Develop study skills and communication skills in various professional situations.
- To study engineering subjects more effectively and critically using the theoretical and practical components of the syllabus.

UNIT - I

Chapter entitled '**Toasted English' by R. K. Narayan** from **"English: Language, Context and Culture"** published by Orient Black Swan, Hyderabad.2022

Vocabulary: The Concept of Word Formation -The Use of Prefixes and Suffixes - Acquaintance with Prefixes and Suffixes from Foreign Languages to form Derivatives - Synonyms and Antonyms

Grammar: Identifying Common Errors in Writing with Reference to Articles and Prepositions.

Reading: Reading and Its Importance- Techniques for Effective Reading.

Writing: Sentence Structures -Use of Phrases and Clauses in Sentences- Simple, Compound & Complex Sentences - Importance of Proper Punctuation- Techniques for Writing precisely – Paragraph Writing – Types, Structures and Features of a Paragraph - Creating Coherence- Organizing Principles of Paragraphs in Documents.

UNIT - II

Chapter entitled **'Appro JRD' by Sudha Murthy** from **"English: Language, Context and Culture" published** by Orient BlackSwan, Hyderabad. 2022. Print.

Vocabulary: Words Often Misspelt - Homophones, Homonyms and Homographs

Grammar: Identifying Common Errors in Writing with Reference to Noun-pronoun Agreement and Subject-verb Agreement, Collocations.

Reading: Sub-Skills of Reading – Skimming and Scanning – Exercises for Practice

Writing: Nature and Style of Writing- Defining /Describing People, Objects, Places and Events – Classifying- Providing Examples or Evidence.

UNIT - III

Chapter entitled **'Lessons from Online Learning' by F. Haider Alvi, Deborah Hurst et al** from **"English: Language, Context and Culture"** published by Orient Black Swan, Hyderabad. 2022. Print.

Vocabulary: Words Often Confused - Words from Foreign Languages and their Use in English.

Grammar: Identifying Common Errors in Writing with Reference to Misplaced Modifiers and tenses

Reading: Sub-Skills of Reading – Intensive Reading and Extensive Reading – Exercises for Practice.

Writing: Format of a Formal Letter-Writing Formal Letters E.g., Letter of Complaint, Letter of Requisition, Email Etiquette, Job Application with CV/Resume.

UNIT - IV

Chapter entitled **'Art and Literature' by Abdul Kalam** from **"English: Language, Context and Culture"** published by Orient Black Swan, Hyderabad. 2022. Print.

Vocabulary: Standard Abbreviations in English, Idioms & Phrasal Verbs.

Grammar: Redundancies and Clichés in Oral and Written Communication. **Reading**: Survey, Question, Read, Recite and Review (SQ3R Method) - Exercises for Practice

Writing: Writing Practices- Essay Writing-Writing Introduction and Conclusion -Précis Writing.

UNIT - V

Chapter entitled **'Go, Kiss the World' by Subroto Bagchi** from **"English: Language, Context and Culture"** published by Orient Black Swan, Hyderabad. 2022. Print.

Vocabulary: Technical Vocabulary and their Usage

Grammar: Common Errors in English (Covering all the other aspects of grammar which were not covered in the previous units)

Reading: Reading Comprehension-Exercises for Practice

Writing: Technical Reports- Introduction – Characteristics of a Report – Categories of Reports Formats- Structure of Reports (Manuscript Format) -Types of Reports - Writing a Report.

Note: Listening and Speaking Skills which are given under Unit-6 in AICTE Model Curriculum are covered in the syllabus of ELCS Lab Course.

Note: 1. As the syllabus of English given in AICTE Model Curriculum-2018 for B. Tech First Year is **Open-ended**, besides following the prescribed textbook, it is required to prepare teaching/learning materials **by the teachers collectively** in the form of handouts based on the needs of the students in their respective colleges for effective teaching/learning in the class.

Note: **2**.Based on the recommendations of NEP2020, teachers are requested to be flexible to adopt Blended Learning in dealing with the course contents .They are advised to teach 40 percent of each topic from the syllabus in blended mode.

Course Outcomes: Students will be able to:

CO-1: Understand the importance of vocabulary and sentence structures.

CO-2: Choose appropriate vocabulary and sentence structures for their oral and written communication.

CO-3: Demonstrate their understanding of the rules of functional grammar.

CO-4: Develop comprehension skills using known and unknown passages.

CO-5: Take an active part in drafting paragraphs, letters, essays, abstracts, précis and reports in various contexts.

TEXT BOOK:

1. ***English: Language, Context and Culture*** by Orient Black Swan Pvt. Ltd, Hyderabad. 2022. Print.

REFERENCE BOOKS:

1. Effective Academic Writing (Second Edition)by Rhonda Liss and Jason Davis Oxford University Press
2. Richards, Jack C. (2022) Interchange Series. Introduction, 1, 2, 3. Cambridge University Press
3. Wood, F.T. (2007). Remedial English Grammar. Macmillan.

4. Chaudhuri, Santanu Sinha. (2018). Learn English: A Fun Book of Functional Language, Grammar and Vocabulary. (2nd ed.,). Sage Publications India Pvt. Ltd.
5. Technical Communication. Wiley India Pvt. Ltd, (2019).
6. Vishwamohan, Aysha. (2013). English for Technical Communication for Engineering Students. Mc Graw-Hill Education India Pvt. Ltd.
7. Swan, Michael.(2016).Practical English Usage. Oxford University Press.4th Edition.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
ELECTRONIC DEVICES AND CIRCUITS

B.Tech I Year II Sem. **L T P C**
2 0 0 2

Course Objectives:

- To familiarize the student with the principle of operation of Junction diode, BJT and FET.
- To know the applications of devices.
- To know the switching characteristics of devices.
- To understand the characteristics and operation of various special purpose devices.

UNIT - I

Diodes: Diode - Static and Dynamic resistances, Equivalent circuit, Diffusion and Transition Capacitances, V-I Characteristics, Diode as a switch- switching times.

UNIT - II

Diode Applications: Rectifier - Half Wave Rectifier, Full Wave Rectifier, Bridge Rectifier, Rectifiers with Capacitive and Inductive Filters, Clippers-Clipping at two independent levels, Clamper-Clamping Circuit Theorem, Clamping Operation, Types of Clampers.

UNIT - III

Bipolar Junction Transistor (BJT): Principle of Operation, Common Emitter, Common Base and Common Collector Configurations, Transistor as a switch, switching times

UNIT - IV

Junction Field Effect Transistor (FET): Construction, Principle of Operation, Pinch-Off Voltage, Volt- Ampere Characteristic, Comparison of BJT and FET, FET as Voltage Variable Resistor, MOSFET, MOSTET as a capacitor.

UNIT – V

Special Purpose Devices: Zener Diode - Characteristics, Zener diode as Voltage Regulator, Principle of Operation - SCR, Tunnel diode, UJT, Varactor Diode, Photo diode, Solar cell, LED, Schottky diode.

Course Outcomes: Upon completion of the Course, the students will be able to:

CO-1: Acquire the knowledge of PN diode and its characteristics.
CO-2: Design the rectifiers with and without filters for specified DC voltage.
CO-3: Illustrate the voltage- current characteristics of Junction Transistor and different configurations of transistor
CO-4: Acquire knowledge about the construction, theory and characteristics of FET and MOSFET
CO-5: Acquire the knowledge about the role of special purpose devices and their applications.

TEXT BOOKS:

1. Jacob Millman - Electronic Devices and Circuits, McGraw Hill Education
2. Electronic Devices and Circuits – R.L. Boylestad and Louis Nashelsky, 9 Ed., 2006, PEI/PHI.

REFERENCE BOOKS:

1. Horowitz -Electronic Devices and Circuits, David A. Bell – 5th Edition, Oxford.
2. Chinmoy Saha, Arindam Halder, Debaati Ganguly - Basic Electronics-Principles and Applications, Cambridge, 2018.
3. Electronic Devices and Circuits – S.Salivahanan, N.Suresh Kumar, A.Vallavaraj, 2 Ed., 2008, TMH.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
APPLIED PHYSICS LABORATORY

B.Tech I Year II Sem. **L T P C**
0 0 3 1.5

Course Objectives:

The objectives of this course for the student to

- Capable of handling instruments related to the Hall effect and photoelectric effect experiments and their measurements.
- Understand the characteristics of various devices such as PN junction diode, Zener diode, BJT, LED, solar cell, lasers and optical fibre and measurement of energy gap and resistivity of semiconductor materials.
- Able to measure the characteristics of dielectric constant of a given material.
- Study the behavior of B-H curve of ferromagnetic materials.
- Able to measure the time Constant of RC Circuit.

LIST OF EXPERIMENTS:

1. Determination of work function and Planck's constant using photoelectric effect.
2. Determination of Hall co-efficient and carrier concentration of a given semiconductor.
3. Characteristics of series and parallel LCR circuits.
4. V-I characteristics of a p-n junction diode and Zener diode.
5. Input and output characteristics of BJT (CE, CB & CC configurations).
6. a) V-I and L-I characteristics of light emitting diode (LED)
   b) V-I Characteristics of solar cell
7. Determination of Energy gap of a semiconductor.
8. Determination of the resistivity of semiconductor by two probe method.
9. Study B-H curve of a magnetic material.
10. Determination of dielectric constant of a given material.
11. a) Determination of the beam divergence of the given LASER beam.
    b) Determination of Acceptance Angle and Numerical Aperture of an optical fiber.
12. Study of Decay Charge and Determination of Time Constant of RC Circuit

Note: Any 8 experiments are to be performed.

Course Outcomes: The students will be able to:

CO-1: Know the determination of the Planck's constant using Photo electric effect and identify the material whether it is n-type or p-type by Hall experiment.
CO-2: Appreciate quantum physics in semiconductor devices and optoelectronics.
CO-3: Gain the knowledge of applications of dielectric constant.
CO-4: Understand the variation of magnetic field and behavior of hysteresis curve.
CO-5: Gain the knowledge of decay of charge and determine time constant of RC circuit

REFERENCE BOOK:

1. S. Balasubramanian, M.N. Srinivasan "A Text book of Practical Physics" S Chand Publishers, 2017.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
PYTHON PROGRAMMING LABORATORY

B.Tech I Year II Sem.

L	T	P	C
0	1	2	2

Course Objectives:

- To install and run the Python interpreter
- To learn control structures.
- To Understand Lists, Dictionaries in python
- To Handle Strings and Files in Python

Lecture- 1: Introduction to Python, Write and Execute a simple python Program, Basic Commands, Variables, Statements, Input /Output, Keywords, Standard Data Types, Strings, Operands and operators.

Lecture-2: Understanding the Decision Control Structures: The if Statement, A Word on Indentation, The if … else Statement, The if … elif … else Statement.

Lecture- 3: Loop Control Statements: The while Loop, The for Loop, Infinite Loops, Nested Loops. The break Statement, The continue Statement.

Lecture- 4: Function Definition and Execution, Scoping, Arguments, Argument Calling by Keywords, Default Arguments, Function Rules, Return Values.

Lecture- 5: Lists: List, Creating List, Updating the Elements of a List, Sorting the List Elements. Storing Different Types of Data in a List, Nested Lists, Nested Lists as Matrices.

Lecture-6: Tuples: Creating and Accessing the Tuple Elements, Basic Operations on Tuples, Functions to Nested Tuples, Inserting, Modifying and Deleting Elements from a Tuple.

Lecture- 7: Exceptions : What is an Exception?, Exception Handling: try..except..else.., try..finally.., Exceptions Nest, Raising Exceptions, Built-In Exceptions.

Lecture- 8: Files: Working with Files and Directories, File Processing, reading from files, writing to files, merging file contents, Controlling File I/O.

Lecture- 9: Python Classes and Objects. Inheritance in Python. Types of inheritance Python.

Lecture- 10: numpy, Plotpy and Scipy libraries of python and their functionalities. Basic GUI programming using these libraries: text labels and buttons.

Week -1: (Lecture- 1)

1. i) Use a web browser to go to the Python website http://python.org. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.

   ii) Start the Python interpreter and type help () to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3. i) Write a program to calculate compound interest when principal, rate and number of periods is given.

   ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read name, address, email and phone number of a person through keyboard and print

the details.

Week - 2: (Lecture- 2 & 3)

1. Print the below triangle using for loop.
   5
   4 4
   3 3 3
   2 2 2 2
   1 1 1 1 1
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

Week - 3: (Lecture- 5)

1. i) Write a program to convert a list and tuple into arrays.
   ii) Write a program to find common values between two arrays.
2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.
3. Write a function called palindrome that takes a string argument and returnsTrue if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.

Week - 4: (Lecture- 4)

1. Write a function called is_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.
2. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.
   i). Write a function called remove_duplicates that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
   ii). The wordlist I provided, words.txt, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.
   iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
   ii) Remove the given word in all the places in a string?
   iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5: (Lecture- 5)

1. i) Write a python program that defines a matrix and prints
   ii) Write a python program to perform addition of two square matrices
   iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.

Week-6: (Lecture- 10)

1. a. Write a function called draw_rectangle that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
   b. Add an attribute named color to your Rectangle objects and modify draw_rectangle so that it uses the color attribute as the fill color.
   c. Write a function called draw_point that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
   d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called draw_circle that draws circles on the canvas.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 7: (Lecture- 7)

1. Use the structure of exception handling for all general purpose exceptions.

Week - 8: (Lecture- 8)

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. Write a function that reads a file file1 and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week - 9: (Lecture- 9)

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR.
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder.
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

Course Outcomes: On successful completion of this course, students will be able to:

CO-1: Develop the application specific codes using python.
CO-2: Understand Strings, Lists, Tuples and Dictionaries in Python.
CO-3: Understand the structure of exception handling for all general purpose exceptions.
CO-4: Verify programs using modular approach, file I/O, Python standard library. Implement Digital Systems using Python.

TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCE BOOKS:

1. Python for Data Science, Dr. Mohd. Abdul Hameed, Wiley Publications - 1st Ed. 2021.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
3. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Think Python, Allen Downey, Green Tea Press
6. Core Python Programming, W. Chun, Pearson
7. Introduction to Python, Kenneth A. Lambert, Cengage

VAAGDEVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

ENGLISH LANGUAGE AND COMMUNICATION SKILLS LABORATORY

B.Tech I Year II Sem. **L T P C**
0 0 2 1

The **English Language and Communication Skills (ELCS) Lab** focuses on the production and practice of sounds of language and familiarizes the students with the use of English in everyday situations both in formal and informal contexts.

Course Objectives:

- To facilitate computer-assisted multi-media instruction enabling individualized and independent language learning
- To sensitize the students to the nuances of English speech sounds, word accent, into nation and rhythm
- To bring about a consistent accent and intelligibility in students' pronunciation of English by providing an opportunity for practice in speaking
- To improve the fluency of students in spoken English and neutralize the impact of dialects.
- To train students to use language appropriately for public speaking, group discussions and interviews

Syllabus: English Language and Communication Skills Lab (ELCS) shall have two parts:

a. **Computer Assisted Language Learning (CALL) Lab**
b. **Interactive Communication Skills (ICS) Lab**

Listening Skills:

Objectives

1 To enable students develop their listening skills so that they may appreciate the role in the LSRW skills approach to language and improve their pronunciation
2 To equip students with necessary training in listening, so that they can comprehend the speech of people of different backgrounds and regions

Students should be given practice in listening to the sounds of the language, to be able to recognize them and find the distinction between different sounds, to be able to mark stress and recognize and use the right intonation in sentences.

- Listening for general content
- Listening to fill up information
- Intensive listening
- Listening for specific information

Speaking Skills:

Objectives

1. To involve students in speaking activities in various contexts
2. To enable students express themselves fluently and appropriately in social and professional contexts.

• Oral practice
• Describing objects/situations/people
• Role play – Individual/Group activities
• Just A Minute (JAM) Session

The following course content is prescribed for the **English Language and Communication Skills Lab**.

Exercise – I CALL Lab:

Understand: Listening Skill- Its importance – Purpose- Process- Types- Barriers- Effective Listening. *Practice*: Introduction to Phonetics – Speech Sounds – Vowels and Consonants – Minimal Pairs-Consonant Clusters- Past Tense Marker and Plural Marker- *Testing Exercises* **ICS Lab**:

Understand: Spoken vs. Written language- Formal and Informal English.

Practice: Ice-Breaking Activity and JAM Session- Situational Dialogues – Greetings – Taking Leave –Introducing Oneself and Others.

Exercise – II CALL Lab:

Understand: Structure of Syllables – Word Stress– Weak Forms and Strong Forms – Stress pattern in sentences – Intonation.

Practice: Basic Rules of Word Accent - Stress Shift - Weak Forms and Strong Forms- Stress pattern in sentences – Intonation - *Testing Exercises*

ICS Lab:

Understand: Features of Good Conversation – Strategies for Effective Communication.

Practice: Situational Dialogues – Role Play- Expressions in Various Situations –Making Requests and Seeking Permissions - Telephone Etiquette.

Exercise - III CALL Lab:

Understand: Errors in Pronunciation-Neutralising Mother Tongue Interference (MTI). *Practice:* Common Indian Variants in Pronunciation – Differences between British and American Pronunciation -*Testing Exercises*

ICS Lab:

Understand: Descriptions- Narrations- Giving Directions and Guidelines – Blog Writing *Practice:* Giving Instructions – Seeking Clarifications – Asking for and Giving Directions – Thanking and Responding – Agreeing and Disagreeing – Seeking and Giving Advice – Making Suggestions.

Exercise – IV CALL Lab:

Understand: Listening for General Details.

Practice: Listening Comprehension Tests - *Testing Exercises*

ICS Lab:

Understand: Public Speaking – Exposure to Structured Talks - Non-verbal Communication- Presentation Skills.

Practice: Making a Short Speech – Extempore- Making a Presentation.

Exercise – V CALL Lab:

Understand: Listening for Specific Details.

Practice: Listening Comprehension Tests -*Testing Exercises*

ICS Lab:

Understand: Introduction to Group Discussion, Interview Skills.

Practice: Group Discussion/Mock Interview.

Minimum Requirement of infrastructural facilities for ELCS Lab:

1. **Computer Assisted Language Learning (CALL) Lab:**

The Computer Assisted Language Learning Lab has to accommodate 30 students with 30 systems, with one Master Console, LAN facility and English language learning software for self- study by students.

System Requirement (Hardware component):

Computer network with LAN facility (minimum 30 systems with multimedia) with the following specifications:

i) Computers with Suitable Configuration
ii) High Fidelity Headphones

2. **Interactive Communication Skills (ICS) Lab:**

The Interactive Communication Skills Lab: A Spacious room with movable chairs and audio-visual aids with a Public Address System, LCD and camcorder.

Source of Material (Master Copy):

- *Exercises in Spoken English. Part 1, 2, 3*. CIEFL and Oxford University Press

Note: Teachers are requested to make use of the master copy and get it tailor-made to suit the contentsof the syllabus.

Suggested Software:

- Cambridge Advanced Learners' English Dictionary with CD.
- Grammar Made Easy by Darling Kindersley.
- Punctuation Made Easy by Darling Kindersley.
- Oxford Advanced Learner's Compass, 10th Edition.
- English in Mind (Series 1-4), Herbert Puchta and Jeff Stranks with Meredith Levy, Cambridge.
- English Pronunciation in Use (Elementary, Intermediate, Advanced) Cambridge UniversityPress.
- English Vocabulary in Use (Elementary, Intermediate, Advanced) Cambridge University Press.
- TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS).
- Digital All
- Orell Digital Language Lab (Licensed Version)

Course Outcomes: Students will be able to:

CO-1: Understand the nuances of English language through audio- visual experience and group activities
CO-2: Neutralize their accent for intelligibility
CO-3: Develop their listening skills so that they may appreciate its role in developing LSRW skills of language and improve their pronunciation.
CO-4: Involve in speaking activities in various contexts.
CO-5: Speak with clarity and confidence which in turn enhance their employability skills

REFERENCE BOOKS:

1. *English Language Communication Skills – Lab Manual cum Workbook.* Cengage Learning India Pvt. Ltd, (2022).
2. Shobha, KN & Rayen, J. Lourdes. (2019). *Communicative English – A workbook.* CambridgeUniversity Press
3. Kumar, Sanjay & Lata, Pushp. (2019). *Communication Skills: A Workbook.* Oxford UniversityPress
4. *ELCS Lab Manual: A Workbook for CALL and ICS Lab Activities.* Orient Black Swan Pvt. Ltd, (2016).
5. Mishra, Veerendra et al. (2020). *English Language Skills: A Practical Approach.* CambridgeUniversity Press.
6. https://www.wix.com//

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
IT WORKSHOP

B.Tech I Year II Sem. **L T P C**
0 0 2 1

Course Objectives:

- The IT Workshop for engineers is a training lab course spread over 60 hours.
- The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, PowerPoint and Publisher.

PC Hardware

Task 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Task 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Task 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Task 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both Windows and Linux. Lab instructors should verify the installation and follow it up with a Viva

Internet & World Wide Web

Task1: **Orientation & Connectivity Boot Camp:** Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

Task 2: Web Browsers, Surfing the Web: Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

Task 3: **Search Engines & Netiquette:** Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

Task 4: Cyber Hygiene: Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

LaTeX and WORD

Task 1 – Word Orientation: The mentor needs to give an overview of LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of LaTeX and MS office or equivalent (FOSS) tool Word as word Processors, Details of the four tasks and features that would be covered in each, Using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

Task 2: Using LaTeX and Word to create a project certificate. Features to be covered:- Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in both LaTeX and Word.

Task 3: Creating project abstract Features to be covered:-Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

Task 4: Creating a Newsletter: Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

Excel

Excel Orientation: The mentor needs to tell the importance of MS office or equivalent (FOSS) tool Excel as a Spreadsheet tool, give the details of the four tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

Task 1: Creating a Scheduler - Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text

Task 2 : Calculating GPA - .Features to be covered:- Cell Referencing, Formulae in excel – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, LOOKUP/VLOOKUP

Task 3: Split cells, freeze panes, group and outline, Sorting, Boolean and logical operators, Conditional formatting

Powerpoint

Task 1: Students will be working on basic power point utilities and tools which help them create basic powerpoint presentations. PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in PowerPoint.

Task 2: Interactive presentations - Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts.

Task 3: Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides.

Course Outcomes:

CO-1: Perform Hardware troubleshooting. Understand Hardware components and inter dependencies

CO-2: Safeguard computer systems from viruses/worms

CO-3: Perform calculations using spreadsheets.

CO-4: Document/ Presentation preparation

REFERENCE BOOKS:

1. Comdex Information Technology course tool kit Vikas Gupta, *WILEY Dreamtech*
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, *WILEY Dreamtech*
3. Introduction to Information Technology, ITL Education Solutions limited, *Pearson Education.*
4. PC Hardware - A Handbook – Kate J. Chase *PHI* (Microsoft)
5. LaTeX Companion – Leslie Lamport, *PHI/Pearson*.
6. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – *CISCO Press, Pearson Education*.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – CISCO Press, *Pearson Education.*

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
DIGITAL ELECTRONICS

B.Tech II Year I Sem. **L T P C**
3 0 0 3

Course Objectives: This course aims at through understanding of binary number system, logic gates, combination logic and synchronous and asynchronous logic.

UNIT - I:
BOOLEAN ALGEBRA AND LOGIC GATES: Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes, Binary Storage and Registers, Binary logic.
Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms, other logic operations, Digital logic gates.

UNIT - II:
GATE – LEVEL MINIMIZATION: The map method, Four-variable map, Five-Variable map, product of sums simplification Don't-care conditions, NAND and NOR implementation other Two-level implementations, Exclusive – Or function.

UNIT - III:
COMBINATIONAL LOGIC: Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.

UNIT - IV:
SEQUENTIAL LOGIC: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, state Reduction and Assignment, Design Procedure**.** Registers, shift Registers, Ripple counters, synchronous counters, other counters.

UNIT - V
MEMORIES AND ASYNCHRONOUS SEQUENTIAL LOGIC: Introduction, Random-Access Memory, Memory Decoding, Error Detection and correction Read-only memory, Programmable logic Array programmable Array logic, Sequential Programmable Devices.
Introduction, Analysis Procedure, Circuits with Latches, Design Procedure, Reduction of state and Flow Tables, Race-Free state Assignment Hazards, Design Example.

Course Outcomes: After learning the contents of this paper the student must be able to

1. Acquire the knowledge on numerical information in different forms and Boolean Algebra theorems for Combinational function minimization
2. Design logic circuits by applying minimization techniques and also able to characterize the various logic families for their AC and DC parameter's.
3. Design and analyze various combination logic circuits and understand the fundamental's of sequential circuits .
4. Design and analyze sequential circuits for various cyclic functions.
5. Acquire the knowledge on concepts of Memories and PLA.

TEXT BOOKS:

1. Digital Design – Third Edition, M. Morris Mano, Pearson Education/PHI.
2. Digital Principles and Applications Albert Paul Malvino Donald P. Leach TATA McGraw Hill Edition.
3. Fundamentals of Logic Design, Roth, 5th Edition, Thomson.

REFERENCE BOOKS:

1. Switching and Finite Automata Theory by Zvi. Kohavi, Tata McGraw Hill.
2. Switching and Logic Design, C.V.S. Rao, Pearson Education
3. Digital Principles and Design – Donald D.Givone, Tata McGraw Hill, Edition.
4. Fundamentals of Digital Logic and Microcomputer Design, 5TH Edition, M. Rafiquzzaman John Wiley.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
DATA STRUCTURES

B.Tech II Year I Sem. **L T P C**
3 0 0 3

Prerequisites: Programming for Problem Solving

Course Objectives

- Exploring basic data structures such as stacks and queues.
- Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
- Introduces sorting and pattern matching algorithms

UNIT - I

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks- Operations, array and linked representations of stacks, stack applications, Queues- operations, array and linked representations.

UNIT - II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.

Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, B- Trees, B+ Trees, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching, Red –Black, Splay Trees.

UNIT - IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.

Sorting: Quick Sort, Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

Course Outcomes:

CO-1: To build the basic knowledge to handle operations like insertions, deletions, searching, and traversing mechanisms in linear data structures.
CO-2: Understand Dictionaries, skip list and Hashing Techniques.
CO-3: Ability to have knowledge on general tree structures, search trees, AVL-trees, B-Trees, B+ Trees, Red-Black Trees, Splay Trees.
CO-4: Apply suitable algorithms for graph traversing and sorting techniques.
CO-5: Implement and know the application of algorithms for pattern matching and Tries.

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2 nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.

2. Data Structures using C – A. S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

REFERENCE BOOK:

1. Data Structures: A Pseudocode Approach with C, 2 nd Edition, R. F. Gilberg and B.A.Forouzan, Cengage Learning.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
COMPUTER ORIENTED STATISTICAL METHODS

B.Tech II Year I Sem. **L T P C**
3 1 0 4

Pre-requisites: Mathematics courses of first year of study.

Course Objectives: To learn
- The theory of Probability, Probability distributions of single and multiple random variables
- The sampling theory, testing of hypothesis and making statistical inferences
- Stochastic process and Markov chains.

UNIT - I: Probability **10 L**

Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Baye's Rule,

Random Variables and Probability Distributions: Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions.

UNIT - II: Expectation and discrete distributions **10 L**

Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's Theorem.

Discrete Probability Distributions: Binomial Distribution, Poisson distribution.

UNIT - III: Continuous and Sampling Distributions **10 L**

Uniform Distribution, Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Normal Approximation to the Binomial Distributions.

Fundamental Sampling Distributions: Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem, t - Distribution, F-Distribution.

UNIT - IV: Sample Estimation & Tests of Hypotheses **10 L**

Introduction, Statistical Inference, Classical Methods of Estimation, Single Sample: Estimating the mean, standard error of a point estimate, prediction interval. Two sample: Estimating the difference between two means, Single sample: Estimating a proportion, Two samples: Estimating the difference between two proportions, Two samples: Estimating the ratio of two variances.

Statistical Hypotheses: General Concepts, Testing a Statistical Hypothesis, Single sample: Tests concerning a single mean, Two samples: tests on two means, One sample: test on a single proportion. Two samples: tests on two proportions, Two- sample tests concerning variances.

UNIT-V: Stochastic Processes and Markov Chains **8L**

Introduction to Stochastic processes- Markov process. Transition Probability, Transition Probability Matrix, First order and Higher order Markov process, n-step transition probabilities, Markov chain, Steady state condition, Markov analysis.

Course outcomes: After learning the contents of this paper the student must be able to

CO-1: Apply the concepts of probability and distributions to case studies.

CO-2: Formulate and solve problems involving random variables and apply statistical methods for analyzing experimental data.
CO-3: Apply concept of estimation and testing of hypothesis to case studies.
CO-4: Correlate the concepts of one unit to the concepts in other units.

TEXT BOOKS:

1. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & Statistics For Engineers & Scientists, 9th Ed. Pearson Publishers.
2. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, Khanna publications.
3. S.D.Sharma, Operations Research, Kedarnath and Ramnath Publishers, Meerut, Delhi.

REFERENCE BOOKS:

1. T.T. Soong, Fundamentals of Probability and Statistics For Engineers, John Wiley & Sons, Ltd, 2004.
2. Sheldon M Ross, Probability and statistics for Engineers and scientists, academic press.
3. Miller and Freund's, Probability and Statistics for Engineers, 8th Edition, Pearson Educations.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
COMPUTER ORGANIZATION AND ARCHITECTURE

B.Tech II Year I Sem. **L T P C**
3 0 0 3

Co-requisite: A Course on "Digital Electronics".

Course Objectives

- The purpose of the course is to introduce principles of computer organization and the basic architectural concepts.
- It begins with basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
- Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors

UNIT - I

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT - II

Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT - III

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT - IV

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT - V

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.

Course Outcomes:

CO-1: Understand the architecture of modern computer and different register transfers and instruction types.

CO-2: Develop a detailed understanding of architecture and functionality of central processing unit.

CO-3: Design arithmetic and logical operations with integer and floating-point operands.

CO-4: Understand of I/O devices communicating with Processing Unit.

CO-5: Understand the characteristics of multi processors, Parallel Processing and its applications.

TEXT BOOK:

1. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI.

REFERENCE BOOKS:

1. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, V th Edition, McGraw Hill.
2. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
3. Structured Computer Organization – Andrew S. Tanenbaum, 4 th Edition, PHI/Pearson.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
OBJECT ORIENTED PROGRAMMING THROUGH JAVA

B.Tech II Year I Sem. **L T P C**
3 0 0 3

Course Objectives

- To Understand the basic object-oriented programming concepts and apply them in problem solving.
- To Illustrate inheritance concepts for reusing the program.
- To Demonstrate multitasking by using multiple threads and event handling
- To Develop data-centric applications using JDBC.
- To Understand the basics of java console and GUI based programming

UNIT - I

Object oriented thinking and Java Basics- Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms. A way of viewing world – Agents, responsibility, messages, methods, History of Java, Java buzzwords, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, parameter passing, recursion, nested and inner classes, exploring string class.

UNIT - II

Inheritance, Packages and Interfaces – Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT - III

Exception handling and Multithreading-- Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. String handling, Exploring java.util. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

UNIT - IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, checkbox groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

UNIT - V

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Course Outcomes:

CO-1: Understand the use of OOP concepts and solve real world problems using OOP techniques.
CO-2: Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.
CO-3: Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.
CO4:Able to gain the knowledge on Event handling and develop web related applications using AWT
CO-5: Design and develop GUI based applications using Applets and Swings.

TEXT BOOKS:

1. Java the complete reference, 7th edition, Herbert schildt, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley & sons.
2. An Introduction to OOP, third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y. Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A. Johnson- Thomson.
5. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.
6. Core Java 2, Vol 2, Advanced Features, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education
7. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
8. Java and Object Orientation, an introduction, John Hunt, second edition, Springer. 9. Maurach's Beginning Java2 JDK 5, SPD.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
DATA STRUCTURES LAB

B.Tech II Year I Sem. **L T P C**
0 0 3 1.5

Prerequisites: A Course on "Programming for problem solving".

Course Objectives:

- It covers various concepts of C programming language
- It introduces searching and sorting algorithms
- It provides an understanding of data structures such as stacks and queues.

List of Experiments:

1. Write a program that uses functions to perform the following operations on singly linked list.:

i) Creation ii) Insertion iii) Deletion iv) Traversal

Write a program that uses functions to perform the following operations on doubly linked list.:

i) Creation ii) Insertion iii) Deletion iv) Traversal

2. Write a program that uses functions to perform the following operations on circular linked list.:

i) Creation ii) Insertion iii) Deletion iv) Traversal

3. Write a program that implement stack (its operations) using

i) Arrays ii) Pointers

4. Write a program that implement Queue (its operations) using

i) Arrays ii) Pointers

5. Write a program that implements the following sorting methods to sort a given list of integers in ascending order

i) Quick sort ii) Heap sort iii) Merge sort

6. Write a program to implement the tree traversal methods(Recursive and Non Recursive).

7. Write a program to implement

i) Binary Search tree ii) B Trees iii) B+ Trees iv) AVL trees v) Red - Black trees

8. Write a program to implement the graph traversal methods.

9. Implement a Pattern matching algorithms using Boyer- Moore, Knuth-Morris-Pratt

Course Outcomes:

CO-1: Design the linear data structures such as List, Stack, Queue and its applications
CO-2: Implement non-linear data structure such as Trees, Graphs and its applications
CO-3: Apply suitable algorithms for sorting Techniques
CO-4: Examine appropriate algorithm for pattern matching.

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.
2. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M. J. Augenstein, PHI/Pearson Education.

REFERENCE BOOK:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B. A. Forouzan, Cengage Learning.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

B.Tech II Year I Sem. **L T P C**
0 0 3 1.5

Course Objectives:

- To write programs using abstract classes.
- To write programs for solving real world problems using the java collection framework.
- To write multithreaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.
- To impart hands-on experience with java programming.

Note:

1. Use LINUX and MySQL for the Lab Experiments. Though not mandatory, encourage the use of the Eclipse platform.

2. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed.

List of Experiments:

1. Use Eclipse or Net bean platform and acquaint yourself with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

3. A) Develop an applet in Java that displays a simple message.
B) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

5. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.

6. Write a Java program for the following: Create a doubly linked list of elements.

Delete a given element from the above list. Display the contents of the list after deletion.

7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in the selected color. Initially, there is no message shown.

8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.

10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).

11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).

12. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.

13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Course Outcomes:

CO-1:Able to solve real world problems using the java collection framework.
CO-2: Able to understand the use of abstract classes.
CO-3:Able to develop multithreaded applications.
CO-4: Able to write GUI programs using AWT, Swing and Event Handling.

REFERENCE BOOKS:

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
DATA VISUALIZATION - R PROGRAMMING/ POWER BI

B.Tech II Year I Sem. **L T P C**
0 0 2 1

Course Objectives:

- Effective use of Business Intelligence (BI) technology (Tableau) to apply data visualization
- To discern patterns and relationships in the data.
- To build Dashboard applications.
- To communicate the results clearly and concisely.
- To be able to work with different formats of data sets.

Lab Problems:

1. Understanding Data, What is data, where to find data, Foundations for building Data Visualizations, Creating Your First visualization?

2. Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts(line, bar charts, Tree maps),Using the Show me panel.

3. Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.

4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, Formatting specific parts of the view.

5. Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.

6. Structuring your data, Sorting and filtering Tableau data, Pivoting Tableau data.

7. Advanced Visualization Tools: Using Filters, Using the Detail panel, using the Size panels, customizing filters, Using and Customizing tooltips, Formatting your data with colors.

8. Creating Dashboards & Storytelling, creating your first dashboard and Story, Design for different displays, adding interactivity to your Dashboard, Distributing & Publishing your Visualization.

9. Tableau file types, publishing to Tableau Online, Sharing your visualizations, printing, and Exporting.

10. Creating custom charts, cyclical data and circular area charts, Dual Axis charts.

Course Outcomes: At the end of the course a student should be able to

CO-1: Understand How to import data into Tableau.
CO-2: Understand Tableau concepts of Dimensions and Measures.
CO-3: Develop Programs and understand how to map Visual Layouts and Graphical Properties.
CO-4: Create a Dashboard that links multiple visualizations.

REFERENCE BOOKS:

1. Microsoft Power BI cookbook, Brett Powell, 2nd edition.
2. R Programming for Data Science by Roger D. Peng (References)
3. The Art of R Programming by Norman Matloff Cengage Learning India.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
GENDER SENSITIZATION LAB

B.Tech II Year I Sem. **L T P C**
0 0 2 0

COURSE DESCRIPTION

This course offers an introduction to Gender Studies, an interdisciplinary field that asks critical questions about the meanings of sex and gender in society. The primary goal of this course is to familiarize students with key issues, questions and debates in Gender Studies, both historical and contemporary. It draws on multiple disciplines – such as literature, history, economics, psychology, sociology, philosophy, political science, anthropology and media studies – to examine cultural assumptions about sex, gender, and sexuality.

This course integrates analysis of current events through student presentations, aiming to increase awareness of contemporary and historical experiences of women, and of the multiple ways that sex and gender interact with race, class, caste, nationality and other social identities. This course also seeks to build an understanding and initiate and strengthen programmes combating gender-based violence and discrimination. The course also features several exercises and reflective activities designed to examine the concepts of gender, gender-based violence, sexuality, and rights. It will further explore the impact of gender-based violence on education, health and development.

Objectives of the Course

- To develop students' sensibility with regard to issues of gender in contemporary India.
- To provide a critical perspective on the socialization of men and women.
- To introduce students to information about some key biological aspects of genders.
- To expose the students to debates on the politics and economics of work.
- To help students reflect critically on gender violence.
- To expose students to more egalitarian interactions between men and women.

Unit-I: UNDERSTANDING GENDER

Introduction: Definition of Gender-Basic Gender Concepts and Terminology-Exploring Attitudes towards Gender-Construction of Gender-Socialization: Making Women, Making Men

- Preparing for Womanhood. Growing up Male. First lessons in Caste.

Unit – II: GENDER ROLES AND RELATIONS

Two or Many? -Struggles with Discrimination-Gender Roles and Relations-Types of Gender Roles-Gender Roles and Relationships Matrix-Missing Women-Sex Selection and Its Consequences-Declining Sex Ratio. Demographic Consequences-Gender Spectrum: Beyond the Binary

Unit – III: GENDER AND LABOUR

Division and Valuation of Labour-Housework: The Invisible Labor- "My Mother doesn't Work." "Share the Load."-Work: Its Politics and Economics -Fact and Fiction. Unrecognized and Unaccounted work.

-Gender Development Issues-Gender, Governance and Sustainable Development-Gender and Human Rights-Gender and Mainstreaming.

Unit – IV: GENDER - BASED VIOLENCE

The Concept of Violence- Types of Gender-based Violence-Gender-based Violence from a Human Rights Perspective-Sexual Harassment: Say No!-Sexual Harassment, not Eve-teasing- Coping with Everyday Harassment- Further Reading: "*Chupulu".*

Domestic Violence: Speaking OutIs Home a Safe Place? -When Women Unite [Film]. Rebuilding Lives. Thinking about Sexual Violence Blaming the Victim-"I Fought for my Life…."

Unit – V: GENDER AND CULTURE

Gender and Film-Gender and Electronic Media-Gender and Advertisement-Gender and Popular Literature- Gender Development Issues-Gender Issues-Gender Sensitive Language-Gender and Popular Literature - Just Relationships: Being Together as Equals

Mary Kom and Onler. Love and Acid just do not Mix. Love Letters. Mothers and Fathers. Rosa Parks- The Brave Heart.

Course Outcomes:

1. Students will have developed a better understanding of important issues related to gender in contemporary India.
2. Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender. This will be achieved through discussion of materials derived from research, facts, everyday life, literature and film.
3. Students will attain a finer grasp of how gender discrimination works in our society and how to counter them. Students will acquire insight into the gendered division of labor and its relation to politics and economics.
4. Students will develop a sense of appreciation of women in all walks of life. Men and women students and professionals will be better equipped to work and live in harmony
5. Through providing accounts of studies and movements as well as the new laws that provide protection and relief to women, the textbook will empower students to understand and respond to gender violence.

Note: **Since it is Interdisciplinary Course, Resource Persons can be drawn from the fields of English Literature or Sociology or Political Science or any other qualified faculty who has expertise in this field from engineering departments.**

- Classes will consist of a combination of activities: dialogue-based lectures, discussions, collaborative learning activities, group work and in-class assignments. Apart from the above prescribed book, Teachers can make use of any authentic materials related to the topics given in the syllabus on "Gender".
- **ESSENTIAL READING:** The Textbook, "Towards a World of Equals: A Bilingual Textbook on Gender" written by A.Suneetha, Uma Bhrugubanda, DuggiralaVasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu published by Telugu Akademi, Telangana Government in 2015.

ASSESSMENT AND GRADING:

- Discussion & Classroom Participation: 20%
- Project/Assignment: 30%
- End Term Exam: 50%

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
DISCRETE MATHEMATICS

B.Tech II Year II Sem.

L	T	P	C
3	0	0	3

Course Objectives:

- Introduces elementary discrete mathematics for computer science and engineering.
- Topics include formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

UNIT - I

Mathematical logic: Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT - II

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

UNIT - III

Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Lattices as Partially Ordered Sets, Boolean Algebra.

UNIT - IV

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutation with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, The Principle of Exclusion.

UNIT - V

Graph Theory: Basic Concepts, Isomorphism and Subgraphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

Course Outcomes:

CO-1: Evaluate the notions of propositions, predicate formulae, Rules of inference
CO-2: describe various types of Relations and Functions.
CO-3: Understand the concepts of Algebraic Systems and Lattices.
CO-4: Apply knowledge of Mathematics, Combinations & Permutations, Binomial Multinomial theorems.
CO-5: Apply graph theory in solving computing problems

TEXT BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-Hill, 1st ed.

2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe l. Mott, Abraham Kandel, Teodore P. Baker, Prentis Hall of India, 2nd ed.

REFERENCE BOOKS:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph.P. Grimald, Pearson education, 5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata McGraw Hill publishing co.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
BUSINESS ECONOMICS AND FINANCIAL ANALYSIS

B.Tech II Year II Sem. **L T P C**
3 0 0 3

Course Objective: To learn the basic business types, impact of the economy on Business and Firms specifically. To analyze the Business from the Financial Perspective.

Unit – I: Introduction to Business and Economics

Business: Structure of Business Firm, Theory of Firm, Types of Business Entities, Limited Liability Companies, Sources of Capital for a Company, Non-Conventional Sources of Finance.

Economics: Significance of Economics, Micro and Macro Economic Concepts, Concepts and Importance of National Income, Inflation, Money Supply and Inflation, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Role of Business Economist, Multidisciplinary nature of Business Economics.

UNIT - II: Demand and Supply Analysis

Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making, Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.

Supply Analysis: Determinants of Supply, Supply Function and Law of Supply.

UNIT - III: Production, Cost, Market Structures & Pricing

Production Analysis: Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale, Different Types of Production Functions.

Cost analysis: Types of Costs, Short run and Long run Cost Functions.

Market Structures: Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, Monopolistic Competition. **Pricing:** Types of Pricing, Product Life Cycle based Pricing, Break Even Analysis, Cost Volume Profit Analysis.

UNIT - IV: Financial Accounting: Accounting concepts and Conventions, Accounting Equation, Double-Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, Preparation of Final Accounts (Simple Problems).

UNIT - V: Financial Ratios Analysis: Concept of Ratio Analysis, Importance and Types of Ratios, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios – Analysis and Interpretation (simple problems).

Course Outcomes:

1. Understand the structure of business, nature, scope and importance of business economics.
2. Know the concept of demand, analysis of demand and how elasticity of demand is used for pricing decisions and to evaluate methods for forecasting demand.
3. Understand the concept of production function, cost analysis and market structures to cope up with the prevailing market competition.
4. Develop the ability to record financial transactions and learn to prepare and interpret financial statements.

5. Enhance skills in analyzing financial statements using various financial ratio analysis techniques to assess the financial health and performance of a business.

TEXT BOOKS:

1. D. D. Chaturvedi, S. L. Gupta, Business Economics - Theory and Applications, International Book House Pvt. Ltd. 2013.
2. Dhanesh K Khatri, Financial Accounting, Tata Mc –Graw Hill, 2011.
3. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, Managerial Economics, 2e, Tata Mc Graw Hill Education Pvt. Ltd. 2012.

REFERENCE BOOKS:

1. Paresh Shah, Financial Accounting for Management 2e, Oxford Press, 2015.
2. S. N. Maheshwari, Sunil K Maheshwari, Sharad K Maheshwari, Financial Accounting, 5e, Vikas Publications, 2013.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
OPERATING SYSTEMS

B.Tech II Year II Sem. **L T P C**
3 0 0 3

Prerequisites:

1. A course on "Computer Programming and Data Structures".
2. A course on "Computer Organization and Architecture".

Course Objectives:

- Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection)
- Introduce the issues to be considered in the design and development of operating system
- Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

UNIT - I

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads

UNIT - II

CPU Scheduling - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

UNIT - III

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors

Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT - IV

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT - V

File System Interface and Operations -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

Course Outcomes:

CO-1: Will be able to control access to a computer and the files that may be shared

CO-2: Demonstrate the knowledge of the components of computers and their respective roles in computing.

CO-3: Ability to recognize and resolve user problems with standard operating environments.

CO-4: Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

CO-5: Ability to change access controls to protect files.

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems- Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Crowley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education
5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
DATABASE MANAGEMENT SYSTEMS

B.Tech II Year II Sem. **L T P C**
3 0 0 3

Prerequisites: A course on "Data Structures".

Course Objectives:

- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

UNIT - I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT - II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views.

Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT - III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and

EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, First, Second, Third normal forms, BCNF, lossless join decomposition, multivalued dependencies, Fourth normal form, Fifth normal form.

UNIT - IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

UNIT - V

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM),

B+ Trees: A Dynamic Index Structure.

Course Outcomes:

1. Describe the fundamental elements of relational database management systems.
2. Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.
3. Convert the ER-model to relational tables, and improve the database design by normalization Techniques.
4. Apply the principles of database transaction management, database recovery and security.
5. Familiar with basic database storage structures and access techniques file and page organizations,indexing methods including B tree, and hashing.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, McGraw hill, V edition.3rd Edition
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
SOFTWARE ENGINEERING

B.Tech II Year II Sem. **L T P C**
3 0 0 3

Course Objectives

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. **A Generic view of process**: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). **Process models**: The waterfall model, Spiral model and Agile methodology

UNIT - II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT - IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT - V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. **Quality Management:** Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

Course Outcomes:

1. Define Software Engineering and list core principles of software engineering and understand various process models
2. Ability to translate end-user requirements into system and software requirements.
3. Implement to design UML and software architecture
4. Apply the testing strategies on different level of implementation.
5. Understand and able to compute quality measures and develop a software quality assurance plan for a software development.

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.

REFERENCE BOOKS:

1. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
2. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
4. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
OPERATING SYSTEMS LAB

B.Tech II Year II Sem. **L T P C**
0 0 2 1

Prerequisites: A course on "Programming for Problem Solving", A course on "Computer Organization and Architecture".

Co-requisite: A course on "Operating Systems".

Course Objectives:

- To provide an understanding of the design aspects of operating system concepts through simulation
- Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

List of Experiments:

1. Write C programs to simulate the following CPU Scheduling algorithms a) FCFS b) SJF c) Round Robin d) priority

2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)

3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.

4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.

5. Write C programs to illustrate the following IPC mechanisms a) Pipes b) FIFOs c) Message Queues
d) Shared Memory

6. Write C programs to simulate the following memory management techniques a) Paging b) Segmentation

7. Write C programs to simulate Page replacement policies a) FCFS b) LRU c) Optimal

Course Outcomes:

CO-1: Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
CO-2: Able to implement C programs using Unix system calls
CO-3: Ability to implement interprocess communication between two processes.
CO-4: Ability to design and solve synchronization problems.

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
2. Advanced programming in the Unix environment, W.R.Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition– 2005, Pearson Education/PHI
2. Operating System - A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
DATABASE MANAGEMENT SYSTEMS LAB

B.Tech II Year II Sem. **L T P C**
0 0 2 1

Co-requisites: "Database Management Systems"

Course Objectives:

- Introduce ER data model, database design and normalization
- Learn SQL basics for data definition and data manipulation

List of Experiments:

1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. A. Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.)

B. Nested, Correlated subqueries

7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

Course Outcomes:

CO-1: Understand and Design the ER Model utilized for developing a database.
CO-2: Acquire skills in using SQL commands for data definition and data manipulation.
CO-3: Design database schema for a given application and apply normalization
CO-4: Develop solutions for database applications using procedures, cursors and triggers..

TEXT BOOKS:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill, 3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGraw Hill, V edition.

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
NODE JS/ REACT JS/ DJANGO

B.Tech II Year II Sem. **L T P C**
0 0 2 1

Prerequisites: Object Oriented Programming through Java, HTML Basics

Course Objectives:

- To implement the static web pages using HTML and do client side validation using JavaScript.
- To design and work with databases using Java
- To develop an end to end application using java full stack.
- To introduce Node JS implementation for server side programming.
- To experiment with single page application development using React.

Exercises:

1. Build a responsive web application for shopping cart with registration, login, catalog and cart pages using CSS3 features, flex and grid.
2. Make the above web application responsive web application using Bootstrap framework.
3. Use JavaScript for doing client – side validation of the pages implemented in experiment 1 and experiment 2.
4. Explore the features of ES6 like arrow functions, callbacks, promises, async/await. Implement an application for reading the weather information from openweathermap.org and display the information in the form of a graph on the web page.
5. Develop a java stand alone application that connects with the database (Oracle / mySql) and perform the CRUD operation on the database tables.
6. Create an xml for the bookstore. Validate the same using both DTD and XSD.
7. Design a controller with servlet that provides the interaction with application developed in experiment 1 and the database created in experiment 5.
8. Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism (Cookies, HTTP Session)
9. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.
10. Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)
11. For the above application create authorized end points using JWT (JSON Web Token).
12. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.
13. Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js
14. Create a TODO application in react with necessary components and deploy it into github.

Course Outcomes:

1. To implement the static web pages using HTML and do client side validation using JavaScript.

2. To design and work with databases using Java.
3. To develop an end to end application using java full stack.
4. To introduce Node JS implementation for server side programming and experiment with single page application development using React.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, Wrox Publications, 2010.
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.

VAAGDEVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

LOGICAL REASONING AND QUANTITATIVE APTITUDE

B.Tech. II Year II Sem. **L T P C**

Course Objectives: Upon completing this course, the students will be able to:

- To improve logical thinking with general applications using mathematical concepts like Sequences, series, number theory and probability.
- It also features students to analyze data interpretation and able of improve their mathematical Skills in various general aspects like coding and decoding, Time and Work puzzles solving blood relations etc.

Unit – I: Logical Reasoning 1. Coding and Decoding 2. Distance and Directions 3. Classifications 4. Odd man out and series 5. Clocks and Calendars etc.

Unit – II: Logical ability 1. Blood relations 2. Seating Arrangements 3. Figure Analysis 4. Puzzles etc.

Unit – III: Number systems 1. LCM and HCF 2. Ratio and proportion 3. Simple interest and compound interest 4. Profit and Loss etc.

Unit – IV: Arithmetic ability 1. Time and work 2. Partnerships 3. Time speed and distance 4. Problems on Trains etc.

Unit – V: Mathematical ability 1. Sequence and series 2. Permutations and combination 3. General probability etc.

Course Outcomes: Upon completion of this course, students will be able to:

1. Improve their logical thinking in terms of general and mathematical concepts.
2. Compete in academic as well as competitive levels through which students are able to solve the real world problems.
3. Analyze the number systems.
4. Make quick decisions to face the critical arithmetic problems.
5. Analyze the mathematical problems

Text Books:

1. A modern approach to verbal and non-verbal reasoning by Dr. R.S. Aggarw
2. Quantitative Aptitude by Abhijit Guha Tata McGraw-Hill Company Limited.

Reference Books:

1. Quantitative Aptitude by P.A. Anand (Wiley)
2. Quantitative Aptitude by Dr. R.S. Agarwal.
3. Objective Arithmetic by S.L. Gulati.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS27) DESIGN AND ANALYSIS OF ALGORITHMS

B.Tech III Year I Sem. **L T P C**
3 1 0 4

Prerequisites:

1. A course on "Computer Programming and Data Structures".
2. A course on "Advanced Data Structures".

Course Objectives:

- Introduces the notations for analysis of the performance of algorithms and the data structure of disjoint sets.
- Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate
- Describes how to evaluate and compare different algorithms using worst-, average-, and best case analysis.
- Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

UNIT - I

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Stassen's matrix multiplication.

UNIT - II

Disjoint Sets: Disjoint set operations, union and find algorithms, Priority Queue-Heaps, Heap sort **Backtracking:** General method, applications, n-queen's problem, sum of subsets problem, graph Coloring, Hamiltonian cycles.

UNIT – III

Dynamic Programming: General method, applications- Optimal binary search tree, 0/1 knapsack problem, All pairs shortest path problem, Traveling salesperson problem, Reliability design.

UNIT - IV

Greedy method: General method, applications-Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected components, Biconnected components.

UNIT - V

Branch and Bound: General method, applications - Traveling salesperson problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP-Hard and NP-Complete classes, Cook's theorem.

Course Outcomes:

1. To develop the ability to analyze the running time and prove the correctness of basic algorithms.
2. To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
3. Choose appropriate data structures and algorithm design methods for a specified application
4. Understand the choice of data structures and the algorithm design methods
5. Able to Describe the classes P, NP, and NPComplete and be able to prove that a certain problem is NP-Complete.

TEXT BOOK:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharan, University press, 1998.

REFERENCE BOOKS:

1. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education.
2. Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, PHI Pvt. Ltd./ Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley and sons.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS28) COMPUTER NETWORKS

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites

1. A course on "Programming for problem solving"
2. A course on "Data Structures"

Course Objectives

- The objective of the course is to equip the students with a general overview of the concepts and fundamentals of computer networks.
- Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

UNIT - I

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless Transmission.

Data link layer: Design issues, framing, Error detection and correction.

UNIT - II

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.

Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing,Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.

UNIT - IV

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT - V

Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.

Course Outcomes:

1. Gain the knowledge of the basics of computer networks and functions of each layer in the OSI and TCP/IP reference model.
2. Familiarity with the essential data link layer protocols, network design and implementation.
3. Design and evaluate subnetting, routing protocols to fulfill networking requirements.
4. Compare various congestion control algorithms and identify appropriate transport layer protocol for real time applications.
5. Analyze the features and operations of TCP/UDP, FTP, HTTP, SMTP,SNMP, etc.

TEXT BOOK:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
2. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS29) DEVOPS

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Pre-Requisites:
1. Software Engineering
2. Software Project Management

Course Objectives:
- Understand the skill sets and high-functioning teams involved in Agile, DevOps and related methods to reach a continuous delivery capability.
- Implement automated system update and DevOps lifecycle.

UNIT-I

Introduction to DevOps:
Introduction,Agile development model, DevOps and ITIL. DevOps process and Continuous Delivery, Release management, Scrum, Kanban, delivery pipeline, identifying bottlenecks.

UNIT-II

Software development models and DevOps:
DevOps Lifecycle for Business Agility, DevOps, and Continuous Testing. DevOps influence on Architecture: Introducing software architecture, The monolithic scenario, Architecture rules of thumb, The separation of concerns, Handling database migrations, Micro services and the data tier, DevOps, architecture, and resilience.

UNIT-III

Introduction to project management:
The need for source code control, the history of source code management, Roles and code, source code management system and migrations, shared authentication, Hosted Git servers, Different Git server implementations, Docker intermission, Gerrit, The pull request model, GitLab.

UNIT-IV

Integrating the system:
Build systems, Jenkins build server, Managing build dependencies, Jenkins plugins, and file system layout, The host server, Build slaves, Software on the host, Triggers, Job chaining and build pipelines, Build servers and infrastructure as code, Building by dependency order, Build phases, Alternative build servers, Collating quality measures.

UNIT-V

Testing Tools and Deployment:

Various types of testing, Automation of testing Pros and cons, Selenium - Introduction, Selenium features, JavaScript testing, Testing backend integration points, Test-driven development, REPL-driven development. Deployment of the system: Deployment systems, Virtualization stacks, code execution at the client, Puppet master and agents, Ansible, Deployment tools: Chef, Salt Stack and Docker.

Course Outcomes:

1. Understand the various components of DevOps environment.
2. Identify Software development models and architectures of DevOps.
3. Use different project management and integration tools.
4. Identify various integration tools and plugins.
5. Understand and choose a suitable testing tool and deployment model for the project.

TEXT BOOKS:

1. Joakim Verona., Practical DevOps, Packt Publishing, 2016.

REFERENCE BOOKS:

1. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications.
2. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS24) QUANTUM COMPUTING
(Professional Elective – I)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Course Objectives

- To introduce the fundamentals of quantum computing
- The problem-solving approach using finite dimensional mathematics

UNIT - I

History of Quantum Computing: Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations

UNIT - II

Background Mathematics: Basics of Linear Algebra, Hilbert space, Probabilities and measurements. Background Physics: Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Background Biology: Basic concepts of Genomics and Proteomics (Central Dogma)

UNIT - III

Qubit: Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.

UNIT - IV

Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm.

UNIT - V

Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation. Quantum Information and Cryptography: Comparison between classical and quantum information theory. Quantum Cryptography, Quantum teleportation

Course Outcomes:

1. Understand basics of quantum computing.
2. Understand physical implementation of Qubit.
3. Understand Quantum algorithms and their implementation.
4. Understand The Impact of Quantum Computing on Cryptography.

TEXT BOOKS:

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge

REFERENCE BOOKS:

1. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS30) ADVANCED COMPUTER ARCHITECTURE
(Professional Elective – I)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites: Computer Organization

Course Objectives:

- To impart the concepts and principles of parallel and advanced computer architectures.
- To develop the design techniques of Scalable and multithreaded Architectures.
- To Apply the concepts and techniques of parallel and advanced computer architectures to design modern computer systems

UNIT - I

Theory of Parallelism, Parallel computer models, The State of Computing, Multiprocessors and Multicomputer, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development tracks, Program and network properties, Conditions of parallelism, Program partitioning and Scheduling, Program flow Mechanisms, System interconnect Architectures.

UNIT - II

Principles of Scalable performance, Performance metrics and measures, Parallel Processing applications, Speed up performance laws, Scalability Analysis and Approaches, Hardware Technologies, Processes and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors

UNIT - III

Shared-Memory Organizations, Sequential and weak consistency models, Pipelining and superscalar techniques, Linear Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline design, superscalar pipeline design.

UNIT - IV

Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers, Message-passing Mechanisms, Multivetor and SIMD computers.

UNIT - V

Vector Processing Principles, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine CM-5.

Course Outcomes:

1. Computational models and Computer Architectures.
2. Concepts of parallel computer models.
3. Describe architectural features of advanced processors.
4. Scalable Architectures, Pipelining, Superscalar processors
5. Concepts of SIMD and CM-5.

TEXT BOOK

1. Advanced Computer Architecture, Kai Hwang, 2nd Edition, Tata McGraw Hill Publishers.

REFERENCE BOOKS:

1. Computer Architecture, J.L. Hennessy and D.A. Patterson, 4th Edition, ELSEVIER.
2. Advanced Computer Architectures, S.G.Shiva, Special Indian edition, CRC, Taylor & Francis.
3. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
4. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.
5. Computer Architecture, B. Parhami, Oxford Univ. Press.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22AI11) DATA ANALYTICS
(Professional Elective – I)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites
1. A course on "Database Management Systems".
2. Knowledge of probability and statistics.

Course Objectives:
- To explore the fundamental concepts of data analytics.
- To learn the principles and methods of statistical analysis
- Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms.
- To understand the various search methods and visualization techniques.

UNIT – I

Data Management: Design Data Architecture and manage the data for analysis, understand various sources of Data like Sensors/Signals/GPS etc. Data Management, Data Quality (noise, outliers, missing values, duplicate data) and Data Processing & Processing.

UNIT – II

Data Analytics: Introduction to Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of Data and Variables, Data Modeling Techniques, Missing Imputations etc. Need for Business Modeling.

UNIT - III

Regression – Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization, and Model Building etc.

Logistic Regression: Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.

UNIT – IV

Object Segmentation: Regression Vs Segmentation – Supervised and Unsupervised Learning, Tree Building – Regression, Classification, Overfitting, Pruning and Complexity, Multiple Decision Trees etc. **Time Series Methods:** Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height, Average Energy etc and Analyze for prediction

UNIT – V

Data Visualization: Pixel-Oriented Visualization Techniques, Geometric Projection Visualization Techniques, Icon-Based Visualization Techniques, Hierarchical Visualization Techniques, Visualizing Complex Data and Relations.

Course Outcomes: After completion of this course students will be able to

1. Understand the impact of data analytics for business decisions and strategy
2. Carry out data analysis/statistical analysis
3. To carry out standard data visualization and formal inference procedures
4. Design Data Architecture
5. Understand various Data Sources

TEXT BOOKS:

1. Student's Handbook for Associate Analytics – II, III.
2. Data Mining Concepts and Techniques, Han, Kamber, 3rd Edition, Morgan Kaufmann Publishers.

REFERENCE BOOKS:

1. Introduction to Data Mining, Tan, Steinbach and Kumar, Addision Wisley, 2006.
2. Data Mining Analysis and Concepts, M. Zaki and W. Meira
3. Mining of Massive Datasets, Jure Leskovec Stanford Univ. Anand Rajaraman Milliway Labs Jeffrey D Ullman Stanford Univ.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS31) IMAGE PROCESSING
(Professional Elective – I)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites

1. Students are expected to have knowledge in linear signals and systems, Fourier Transform, basic linear algebra, basic probability theory and basic programming techniques; knowledge of digital signal processing is desirable.
2. A course on "Computational Mathematics"
3. A course on "Computer Oriented Statistical Methods"

Course Objectives

- Provide a theoretical and mathematical foundation of fundamental Digital Image Processing concepts.
- The topics include image acquisition; sampling and quantization; preprocessing; enhancement; restoration; segmentation; and compression.

UNIT - I

Digital Image Fundamentals: Digital Image through Scanner, Digital Camera. Concept of Gray Levels. Gray Level to Binary Image Conversion. Sampling and Quantization. Relationship between Pixels. Imaging Geometry. 2D Transformations-DFT, DCT, KLT and SVD.

UNIT - II

Image Enhancement in Spatial Domain Point Processing, Histogram Processing, Spatial Filtering, Enhancement in Frequency Domain, Image Smoothing, Image Sharpening.

UNIT - III

Image Restoration Degradation Model, Algebraic Approach to Restoration, Inverse Filtering, Least Mean Square Filters, Constrained Least Squares Restoration, Interactive Restoration.

UNIT - IV

Image Segmentation Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region Oriented Segmentation.

UNIT - V

Image Compression Redundancies and their Removal Methods, Fidelity Criteria, Image Compression Models, Source Encoder and Decoder, Error Free Compression, Lossy Compression.

Course Outcomes:

1. Demonstrate the knowledge of the basic concepts of two-dimensional signal acquisition, sampling, and quantization.
2. Demonstrate the knowledge of filtering techniques.
3. Demonstrate the knowledge of 2D transformation techniques.
4. Demonstrate the knowledge of image enhancement, segmentation, restoration and compression techniques.
5. Demonstrate the knowledge of Image compression redundancies and present their removal methods.

TEXT BOOK:

1. Digital Image Processing: R.C. Gonzalez & R. E. Woods, Addison Wesley/ Pearson Education, 2nd Ed, 2004.

REFERENCE BOOKS:

1. Fundamentals of Digital Image Processing: A. K. Jain, PHI.
2. Digital Image Processing using MAT LAB: Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins: Pearson Education India, 2004.
3. Digital Image Processing: William K. Pratt, John Wiley, 3rd Edition, 2004.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS32) PRINCIPLES OF PROGRAMMING LANGUAGES
(Professional Elective – I)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites

1. A course on "Mathematical Foundations of Computer Science".
2. A course on "Computer Programming and Data Structures".

Course Objectives

- Introduce important paradigms of programming languages
- To provide conceptual understanding of high-level language design and implementation
- Topics include programming paradigms; syntax and semantics; data types, expressions and statements; subprograms and blocks; abstract data types; concurrency; functional and logic programming languages; and scripting languages

UNIT - I

Preliminary Concepts: Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments Syntax and Semantics: General Problem of Describing Syntax and Semantics, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meanings of Programs

UNIT - II

Names, Bindings, and Scopes: Introduction, Names, Variables, Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants

Data Types: Introduction, Primitive Data Types, Character String Types, User Defined Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence Expressions and Statements, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode

Assignment Control Structures – Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

UNIT - III

Subprograms and Blocks: Fundamentals of Sub-Programs, Design Issues for

Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that Are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User Defined Overloaded Operators, Closures, **Coroutines Implementing Subprograms:** General Semantics of Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping

Abstract Data Types: The Concept of Abstraction, Introductions to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations

UNIT - IV

Concurrency: Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Java Threads, Concurrency in Function Languages, Statement Level Concurrency.

Exception Handling and Event Handling: Introduction, Exception Handling in Ada, C++, Java, Introduction to Event Handling, Event Handling with Java and C#.

UNIT - V

Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages

Logic Programming Language: Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming.

Scripting Language: Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library. (Text Book 2)

Course Outcomes:

1. Analyze Syntax related concepts including context free grammars, Attribute Grammar parse trees.
2. Perceive the semantic issues associated with function implementations.
3. Perceive the concepts of Abstraction and Encapsulation constructs of classes ,interfaces, packages of various Language Examples.
4. Perceive the implementation of object oriented languages.
5. Compare the Functional Programming Languages and Logic Programming Languages

TEXT BOOKS:

1. Concepts of Programming Languages Robert. W. Sebesta 10/E, Pearson Education.
2. Programming Language Design Concepts, D. A. Watt, Wiley Dreamtech, 2007.

REFERENCE BOOKS:

1. Programming Languages, 2nd Edition, A.B. Tucker, R. E. Noonan, TMH.
2. Programming Languages, K. C. Louden, 2nd Edition, Thomson, 2003.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS33) COMPUTER GRAPHICS
(Professional Elective – II)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites

1. Programming for problem solving and Data Structures

Course Objectives:

- Provide the basics of graphics systems including Points and lines, line drawing algorithms, 2D, 3D objective transformations

UNIT – I

Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random-scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms (DDA and Bresenham's Algorithm) circle- generating algorithms and ellipse - generating algorithms

Polygon Filling: Scan-line algorithm, boundary-fill and flood-fill algorithms

UNIT – II

2-D geometric transformations: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, clipping operations, point clipping, Line clipping-Cohen Sutherland algorithms, Polygon clipping-Sutherland Hodgeman polygon clipping algorithm.

UNIT – III

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces, Polygon rendering methods, color models and color applications.

UNIT – IV

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

viewing: Viewing pipeline, viewing coordinates, projections, view volume and general projection
transforms and clipping.

UNIT – V

Computer animation: Design of animation sequence, general computer animation functions, raster animations, computer animation languages, key frame systems, motion specifications.
Visible surface detection methods: Classification, back-face detection, depth-buffer method, BSP- tree method, area sub-division method and octree method.

Course Outcomes:

1. Discover the applications of computer graphics
2. Understand 2D geometric transformations and clipping algorithms
3. Understand 3D object representations, curves, surfaces, polygon rendering methods, color models
4. Understand 3D geometric transformations, viewing and clipping
5. Analyze animation sequence and visible surface detection methods

TEXT BOOKS:

1. "Computer Graphics C version", Donald Hearn and M. Pauline Baker, Pearson Education

REFERENCE BOOKS:

1. Procedural elements for Computer Graphics, David F Rogers, Tata Mc Graw hill, 2nd edition.
2. Principles of Interactive Computer Graphics", Neuman and Sproul, TMH.
3. Principles of Computer Graphics, Shalini Govil, Pai, 2005, Springer.
4. "Computer Graphics Principles & practice", second edition in C, Foley, Van Dam, Feiner and Hughes, Pearson Education.
5. Computer Graphics, Steven Harrington, TMH.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS34) EMBEDDED SYSTEMS
(Professional Elective – II)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Pre-requisites:
- A course on "Digital Logic Design and Microprocessors"
- A course on "Computer Organization and Architecture"

Course Objectives:

- To provide an overview of principles of Embedded System
- To provide a clear understanding of role of firmware, operating systems in correlation with hardware systems.

UNIT - I

Introduction to Embedded Systems: Processor embedded into a system, Embedded Hardware units and devices in a system, Embedded software in a system, Design process of an embedded system, classification of embedded systems, characteristics and quality attributes of an embedded systems

UNIT - II

Introduction to processor/microcontroller architecture, Real world interfacing, processor and memory organization, memory types, memory maps and addresses, interrupt sources and interrupt service mechanism.

UNIT - III

On board Communication Basics: serial; communication devices, Parallel devices, Wireless devices, Real time clock, Serial bus communication Protocols - I2C, SPI; Parallel bus communication - ISA, PCI.

UNIT - IV

Embedded Firmware Development: Overview of programming concepts - in assembly language and in high level language 'C', C Program elements- Heads, Source files, Processor Directives, Macros, Functions, Data types and Data Structures

UNIT - V

OS Based Embedded Systems: OS services - Process/Task Management, Memory Management, I/O subsystem manager, Inter Process/Task communications - Tasks, Task states, Shared data, Signals, Message Queues, Mailbox, Pipes and concepts of Semaphores.

Course Outcomes:

1. Expected to understand the selection procedure of processors in the embedded domain.
2. Understanding the concepts of Microcontrooler architecture
3. Concepts of Serial/Parallel bus communication protocols
4. Design procedure of embedded firm ware.
5. Understand the concepts of Interprocess Communication

TEXT BOOK:

1. Embedded Systems, Raj Kamal, 2nd edition, Tata Mc Graw Hill
2. Shibu K V, "Introduction to Embedded Systems", Second Edition, Mc Graw Hill

REFERENCE BOOKS:

1. Rajkamal, Embedded Systems Architecture, Programming and Design, Tata McGraw-Hill
2. Frank Vahid and Tony Givargis, "Embedded Systems Design" - A Unified Hardware/Software Introduction, John Wiley
3. Lyla, "Embedded Systems" –Pearson
4. David E. Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint 2000.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS35) INFORMATION RETRIEVAL SYSTEMS
(Professional Elective – II)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites:
1. Data Structures.

Course Objectives:

- To learn the concepts and algorithms in Information Retrieval Systems
- To understand the data/file structures that are necessary to design, and implement information retrieval (IR) systems.

UNIT - I
Introduction to Information Retrieval Systems: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses Information Retrieval System Capabilities: Search Capabilities, Browse Capabilities, Miscellaneous Capabilities

UNIT - II
Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models.

UNIT - III
Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages
Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

UNIT - IV
User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies

UNIT - V

Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems

Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval

Course Outcomes:

1. Ability to apply IR principles to locate relevant information, large collections of data and Data Warehouses Information Retrieval System Capabilities.
2. Ability to design different Cataloging and Indexing.
3. Ability to design different document clustering algorithms.
4. Implement retrieval systems for User Search Techniques.
5. Design an Information Retrieval System for web search tasks.

TEXT BOOK:

1. Information Storage and Retrieval Systems – Theory and Implementation, Second Edition, Gerald J. Kowalski, Mark T. Maybury, Springer

REFERENCE BOOKS:

1. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
2. Information Storage & Retrieval by Robert Korfhage – John Wiley & Sons.
3. Modern Information Retrieval by Yates and Neto Pearson Education.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS36) DISTRIBUTED DATABASES
(Professional Elective – II)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites:
1. A course on "Database Management Systems"

Course Objectives:
- The purpose of the course is to enrich the previous knowledge of database systems and expose the need for distributed database technology to confront the deficiencies of the centralized database systems.
- Introduce basic principles and implementation techniques of distributed database systems.
- Equip students with principles and knowledge of parallel and object-oriented databases.
- Topics include distributed DBMS architecture and design; query processing and optimization; distributed transaction management and reliability; parallel and object database management systems.

UNIT - I
Introduction; Distributed Data Processing, Distributed Database System, Promises of DDBSs, Problem areas.
Distributed DBMS Architecture: Architectural Models for Distributed DBMS, DDBMS Architecture. **Distributed Database Design**: Alternative Design Strategies, Distribution Design issues, Fragmentation, Allocation.

UNIT - II
Query processing and decomposition: Query processing objectives, characterization of query processors, layers of query processing, query decomposition, localization of distributed data.
Distributed query Optimization: Query optimization, centralized query optimization, distributed query optimization algorithms.

UNIT - III
Transaction Management: Definition, properties of transaction, types of transactions, distributed concurrency control: serializability, concurrency control mechanisms & algorithms, time - stamped & optimistic concurrency control Algorithms, deadlock Management.

UNIT - IV
Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance

in distributed systems, failures in Distributed DBMS, local & distributed reliability protocols, site failures and network partitioning.

Parallel Database Systems: Parallel database system architectures, parallel data placement, parallel query processing, load balancing, database clusters.

UNIT - V

Distributed object Database Management Systems: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query Processing.

Object Oriented Data Model: Inheritance, object identity, persistent programming languages, persistence of objects, comparison OODBMS and ORDBMS

Course Outcomes:

1. Understand distributed database systems architecture and design.
2. Be able to apply methods and techniques for distributed query processing and optimisation.
3. Understand the broad concepts of distributed transaction process.
4. Study and identify various issues related to the development of distributed database systems.
5. Understand the design aspects of object-oriented database systems and related developments.

TEXT BOOKS:

1. M. Tamer OZSU and Patuck Valduriez: Principles of Distributed Database Systems, Pearson Edn. Asia, 2001.
2. Stefano Ceri and Giuseppe Pelagatti: Distributed Databases, McGraw Hill.

REFERENCE BOOK:

1. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: “Database Systems: The Complete Book”, Second Edition, Pearson International Edition.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22AI12) NATURAL LANGUAGE PROCESSING
(Professional Elective – II)

B.Tech III Year I Sem. **L T P C**
3 0 0 3

Prerequisites:
1. Data structures and compiler design

Course Objectives:
- Introduction to some of the problems and solutions of NLP and their relation to linguistics and statistics.

UNIT – I

Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models
Finding the Structure of Documents: Introduction, Methods, Complexity of the Approaches, Performances of the Approaches, Features

UNIT – II

Syntax I: Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms

UNIT – III

Syntax II: Models for Ambiguity Resolution in Parsing, Multilingual Issues
Semantic Parsing I: Introduction, Semantic Interpretation, System Paradigms, Word Sense

UNIT – IV

Semantic Parsing II: Predicate-Argument Structure, Meaning Representation Systems

UNIT – V

Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Bayesian parameter estimation, Language Model Adaptation, Language Models-class based, variable length, Bayesian topic based, Multilingual and Cross Lingual Language Modeling

Course Outcomes:

1. Understand the concept of natural language processing.
2. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
3. Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems
4. Manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
5. Design, implement, and analyze NLP algorithms; and design different language modeling Techniques.

TEXT BOOKS:

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.

REFERENCE BOOK:

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications.
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS37) COMPUTER NETWORKS LAB

B.Tech III Year I Sem. **L T P C**
0 0 2 1

Course Objectives

- To understand the working principle of various communication protocols.
- To understand the network simulator environment and visualize a network topology and observe its performance
- To analyze the traffic flow and the contents of protocol frames

List of Experiments

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting techniques used in buffers.
10. **Wire shark**
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics

    & Filters. How to run Nmap
    scan
    Operating System Detection
    using Nmap Do the following
    using NS2 Simulator
    i. NS2 Simulator-Introduction
    ii. Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv. Simulate to Find the Number of Packets Dropped due to Congestion
    v. Simulate to Compare Data Rate & Throughput.
    vi. Simulate to Plot Congestion for Different Source/Destination

vii. Simulate to Determine the Performance with respect to Transmission of Packets

Course Outcomes:

1. Implement data link layer farming methods
2. Analyze error detection and error correction codes.
3. Implement and analyze routing and congestion issues in network design.
4. Implement Encoding and Decoding techniques used in presentation layer

TEXT BOOK:

1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI.

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education.
2. Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS38) DEVOPS LAB

B.Tech III Year I Sem. **L T P C**
0 0 2 1

Course Objectives:

- Develop a sustainable infrastructure for applications and ensure high scalability. DevOps aims to shorten the software development lifecycle to provide continuous delivery with high-quality.

List of Experiments:

1. Write code for a simple user registration form for an event.
2. Explore Git and GitHub commands.
3. Practice Source code management on GitHub. Experiment with the source code in exercise 1.
4. Jenkins installation and setup, explore the environment.
5. Demonstrate continuous integration and development using Jenkins.
6. Explore Docker commands for content management.
7. Develop a simple containerized application using Docker.
8. Integrate Kubernetes and Docker
9. Automate the process of running containerized application for exercise 7 using Kubernetes.F
10. Install and Explore Selenium for automated testing.
11. Write a simple program in JavaScript and perform testing using Selenium.
12. Develop test cases for the above containerized application using selenium.

Course Outcomes:

1. Understand the need of DevOps tools
2. Understand the environment for a software application development
3. Apply different project management, integration and development tools
4. Use Selenium tool for automated testing of application

TEXT BOOKS:

1. Joakim Verona., Practical DevOps, Packt Publishing, 2016.

REFERENCE BOOKS:

1. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications.
2. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22EN03) ADVANCED ENGLISH COMMUNICATION SKILLS LAB

B.Tech III Year I Sem. **L T P C**
0 0 2 1

Prerequisite: English Language and Communication Skills Lab

Introduction

The introduction of the Advanced English Communication Skills Lab is considered essential at 3rd year level. At this stage, the students need to prepare themselves for their careers which may require them to listen, speak, read, and write in English both for their professional and interpersonal communication in the globalised context.

The proposed lab should be a laboratory course to enable students to use 'good' English and perform the following:

- Gather ideas and information to organize ideas relevantly and coherently.
- Engage in debates.
- Participate in group discussions.
- Face interviews.
- Write project/research reports/technical reports.
- Make oral presentations.
- Write formal letters.
- Transfer information from non-verbal to verbal texts and vice-versa.
- Take part in social and professional communication.

Course Objectives:

This Lab focuses on using multi-media instruction for language development to meet the following targets. By the end of the course the students will be able

- To prepare the students for their placements by improving communication skills.
- To familiarize and train students with the types and elements of Résumé/ Curriculum Vitae.
- To expose students to the concepts of report writing familiarize and train students to write technical reports.
- To practice interview skills as an observer, an interviewer and/or an interviewee.

1. Syllabus

Module 1: Fundamentals of Interpersonal Communication

- Listen to process information- give information, as part of a simple explanation – conversation starters: small talk-exposure to functional aspects of intonation- accent- tone-

pauses for practice – compare information and ideas from multiple sources- converse with reasonable accuracy over a wide range of everyday topics.

- Lexical chunks for accuracy and fluency- factors influencing the fluency, Role play-deliver a five-minute formal / informal talk – greetings – respond to greetings – invite and offer – accept – decline – take leave- making a request-apology etc.
- Listening for gist- listening for details-Being an active listener: giving verbal and non-verbal feedback – Summarizing academic readings and lectures - conversational speech listening to and participating in conversation – persuasion.

Module 2: Effective Writing Skills

- Résumé Writing-Concept of Résumé Writing-Professional career objective-Resume-Curriculum vitae, Biodata: Difference-Format of Résumé and Types of Résumés -Tips to build a winning Résumé-Tips to write effective cover letter-Statement of Purpose-Letters of Recommendation.
- Report Writing-Elements of Report Writing- Significance, format, layout, and mechanism-types of Reports-Newspaper Reports-Technical reports -Special Reports-Report in manuscript format.

Module 3: Presentation Skills

Oral presentations - individual and group through JAMsessions/seminars/PPTs and Written presentations through posters/projects/reports.

Module 4: Group Discussion

Concept and process of Group Discussion-Importance of Group Discussion-Do's and Don'ts of Group Discussion-Group Discussion for placements-Practice on topics–Current affairs, Abstract topics, General awareness, Business and economy, Education and Social issues.

Module 5: Interview Skills

Interview Skills: Meaning and Purpose of an Interview-Types of interviews-telephonic interview, video conference-(n)etiquette; Interview preparation techniques-Dress code at an interview-Types of interview questions-FAQs in HR Interview.

Minimum Requirement of infrastructural facilities for Advanced English Communication Skills Lab:

The Advanced English Communication Skills (AECS) Laboratory shall have the following infrastructural facilities to accommodate at least 35 students in the lab:

- Spacious room with appropriate acoustics

- Audio-visual aids
- LCD Projector
- Public Address system
- P – IV Processor, Hard Disk – 80 GB, RAM–512 MB Minimum, Speed – 2.8 GHZ
- T. V, a digital stereo
- Headphones of High quality
- Forty movable chairs and Eight round tables to accommodate 5 students per each table

Course Outcomes:

After completing this course, students will be able to:

1. Participate in group discussion to present their viewpoints briefly and effectively.
2. Inculcate flair for writing and felicity in written expression in Résumé / Curriculum Vitae / Reports.
3. Participate confidently with appropriate body language in interviews.
4. Enhance their team building skills and capabilities for effective decision making.

Prescribed Lab Manual: A book titled ***A Coursebook of Advanced Communication Skills Lab*** published by Universities Press, Hyderabad.

Suggested Software: The software of the topics prescribed above are procured and used.

- **Globarena**
- **Open source software**
- **Oxford Advanced Learner's Compass**, 8th Edition
- **DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.**
- **Lingua TOEFL CBT Insider**, by Dreamtech
- **TOEFL & GRE** (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
- **The following software from 'train2success.com'**
  - **Positive Thinking**
  - **Interviewing Skills**
  - **Telephone Skills**
  - **Time Management**
  - **Skill mate**
  - **Presentation skills, Cambridge** (with VCD)

Books Prescribed:

1. **Technical Communication** by Meenakshi Raman & Sangeeta Sharma, Oxford University Press 2009.

2. **English Language Communication : A Reader cum Lab Manual** Dr A Ramakrishna Rao, Dr G Natanam& Prof SA Sankaranarayanan, Anuradha Publications, Chennai 2008.
3. **Advanced Communication Skills Laboratory Manual** by Sudha Rani, D, Pearson Education 2011.
4. **Technical Communication** by Paul V. Anderson. 2007. Cengage Learning pvt. Ltd. New Delhi.
5. **Business and Professional Communication:** Keys for Workplace Excellence. Kelly M. Quintanilla & Shawn T. Wahl. Sage South Asia Edition. Sage Publications. 2011.

DISTRIBUTION AND WEIGHTAGE OF MARKS:

- The practical examinations for the Laboratory practice shall be conducted as per the University norms prescribed for the core engineering practical sessions.

- For the English Language lab sessions, there shall be continuous evaluation during the year for 30 sessional marks and 70 End Examination marks. Of the 30 marks, 20 marks shall be awarded for day-to-day work and 10 marks to be awarded by conducting Internal Lab Test(s). The End Examination shall be conducted by the teacher concerned, by inviting the External Examiner from outside. In case of the non-availability of the External Examiner, other teacher of the same department can act as the External Examiner.

Mini Project: As a part of Internal Evaluation

- **Seminar/ Professional Presentation**
- **A Report on the same has to be prepared and presented.**
- ***Teachers may use their discretion to choose topics relevant and suitable to the needs of students.***
- ***Not more than two students to work on each mini project.***
- ***Students may be assessed by their performance both in oral presentation and written report.***

****

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS39) UI DESIGN-FLUTTER

B.Tech III Year I Sem. **L T P C**
0 0 2 1

Course Objectives:

0. To learn installation of SDK of Flutter, Xcode and Android Emulator
1. Understanding Stateless and Stateful Widgets and Widget Tree
2. Learning of Dart basics
3. Application of Animation to app.

UNIT - I Introducing Flutter and Getting Started

Introducing Flutter, Defining Widgets and Elements, Understanding Widget Lifecycle Events, The Stateless Widget Lifecycle, The Stateful Widget Lifecycle, Understanding the Widget Tree and the Element Tree, Stateless Widget and Element Trees, Stateful Widget and Element Trees, Installing the Flutter SDK, Installing on mac OS, System Requirements, Get the Flutter SDK, Check for Dependencies, iOS Setup: Install Xcode, Android Setup: Install Android Studio, Set Up the Android Emulator, Installing on Windows, System Requirements, Get the Flutter SDK, Check for Dependencies, Install Android Studio, Set Up the Android Emulator, Installing on Linux, System Requirements, Get the Flutter SDK, Check for Dependencies, Install Android Studio, Set Up the Android Emulator, Configuring the Android Studio Editor.

UNIT - II Creating a Hello World App

Setting Up the Project, Using Hot Reload, Using Themes to Style Your App, Using a Global App Theme, Using a Theme for Part of an App, Understanding Stateless and Stateful Widgets, Using External Packages, Searching for Packages, Using Packages

UNIT - III Learning Dart Basics

Use of Dart, Commenting Code, Running the main() Entry Point, Referencing Variables, Declaring Variables, Numbers, Strings, Booleans, Lists, Maps, Runes, Using Operators, Using Flow Statements, if and else, ternary operator, for Loops, while and do-while, while and break, continue, switch and case, Using Functions, Import Packages, Using Classes, Class Inheritance, Class Mixins, Implementing Asynchronous Programming.

UNIT- IV Creating a Starter Project Template

Creating and Organizing Folders and Files, Structuring Widgets.

Understanding the Widget Tree

Introduction to Widgets, Building the Full Widget Tree, Building a Shallow Widget

Tree, Refactoring with a Constant, Refactoring with a Method, Refactoring with a Widget Class.

UNIT- V Using Common Widgets

Using Basic Widgets, Safe Area, Container, Text, Rich Text, Column, Row, Column and Row Nesting, Buttons, Floating Action Button, Flat Button, Raised Button, Icon Button, Popup Menu Button, Button Bar, Using Images and Icons, Asset Bundle, Image, Icon, Using Decorators, Using the Form Widget to Validate Text Fields, Checking Orientation.

Adding Animation to an App

Using Animated Container, Using Animated Cross Fade, Using Animated Opacity, Using Animation Controlle, Using Staggered Animations,

Course Outcomes:

1. Knowledge on installation of various softwares.
2. Install Flutter in Android Studio.
3. Understanding of various Widgets.
4. Develop Flutter applications using Dart packages.
5. Application of Animation to Apps

TEXT BOOK:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development, 1st edition, Wrox publisher.

REFERENCE BOOKS:

1. Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2, Packt Publishing Limited.
2. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, 1st edition, Apress.
3. Frank Zammetti, Practical Flutter: Improve your Mobile Development with Google's Latest Open-Source SDK, 1st edition, Apress.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22MB06) INTELLECTUAL PROPERTY RIGHTS

B.Tech III Year I Sem. **L T P C**
3 0 0 0

Course Objectives:

- Significance of intellectual property and its protection
- Introduce various forms of intellectual property

UNIT – I

Introduction to Intellectual property: Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

UNIT – II

Trade Marks: Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

UNIT – III

Law of copyrights: Fundamental of copyright law, originality of material, rights of reproduction, rights to perform the work publicly, copyright ownership issues, copyright registration, notice of copyright, International copyright law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT – IV

Trade Secrets: Trade secret law, determination of trade secret status, liability for misappropriations of trade secrets, protection for submission, trade secret litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

UNIT – V

New development of intellectual property: new developments in trade mark law; copyright law, patent law, intellectual property audits.

International overview on intellectual property, international – trade mark law, copyright law, international patent law, and international development in trade secrets law.

Course Outcomes:

1. Distinguish and explain various forms of IPRs.
2. Explains trademarks function and Registration of process
3. Identify criteria to fit one's own intellectual work in particular form of IPRs.
4. Apply statutory provisions to protect particular form of IPRs.
5. Appraise new developments in IPR laws at national and international level

TEXT BOOK:

1. Intellectual property right, Deborah. E. Bouchoux, Cengage learning.

REFERENCE BOOK:

1. Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company ltd.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22AI05) MACHINE LEARNING

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Course Objectives:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To have a thorough understanding of the Supervised and Unsupervised learning techniques
- To study the various probability-based learning techniques

UNIT - I

Learning – Types of Machine Learning – Supervised Learning – The Brain and the Neuron – Design a Learning System – Perspectives and Issues in Machine Learning – Concept Learning Task – Concept Learning as Search – Finding a Maximally Specific Hypothesis – Version Spaces and the Candidate Elimination Algorithm – Linear Discriminants: – Perceptron – Linear Separability – Linear Regression.

UNIT - II

Multi-layer Perceptron– Going Forwards – Going Backwards: Back Propagation Error – Multi-layer Perceptron in Practice – Examples of using the MLP – Overview – Deriving Back-Propagation – Radial Basis Functions and Splines – Concepts – RBF Network – Curse of Dimensionality – Interpolations and Basis Functions – Support Vector Machines

UNIT - III

Learning with Trees – Decision Trees – Constructing Decision Trees – Classification and Regression Trees – Ensemble Learning – Boosting – Bagging – Different ways to Combine Classifiers – Basic Statistics – Gaussian Mixture Models – Nearest Neighbor Methods – Unsupervised Learning – K means Algorithms

UNIT - IV

Dimensionality Reduction – Linear Discriminant Analysis – Principal Component Analysis – Factor Analysis – Independent Component Analysis – Locally Linear Embedding – Isomap – Least Squares Optimization

Evolutionary Learning – Genetic algorithms – Genetic Offspring: - Genetic Operators – Using Genetic Algorithms

UNIT - V

Reinforcement Learning – Overview – Getting Lost Example

Markov Chain Monte Carlo Methods – Sampling – Proposal Distribution – Markov Chain

Monte Carlo
– Graphical Models – Bayesian Networks – Markov Random Fields – Hidden Markov Models – Tracking Methods

Course Outcomes:

1. Learn the basics of learning problems with hypothesis and version spaces
2. Analyze the concept of neural networks for learning linear and non-linear activation functions
3. Understand the principles of evolutionary computing algorithms
4. Understand the fundamental concepts of Genetic Algorithm and Analyze and design the genetic algorithms for optimization engineering problems
5. Design an ensembler to increase the classification accuracy

TEXT BOOKS:

1. Stephen Marsland, ―Machine Learning – An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.

REFERENCE BOOKS:

1. Tom M Mitchell, ―Machine Learning, First Edition, McGraw Hill Education, 2013.
2. Peter Flach, ―Machine Learning: The Art and Science of Algorithms that Make Sense of Data‖, First Edition, Cambridge University Press, 2012.
3. Jason Bell, ―Machine learning – Hands on for Developers and Technical Professionals‖, First Edition, Wiley, 2014
4. Ethem Alpaydin, ―Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series), Third Edition, MIT Press, 2014

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS40) FORMAL LANGUAGES AND AUTOMATA THEORY

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Course Objectives

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability.

UNIT - I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA, Moore and Melay machines

UNIT - II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT - III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a

PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

UNIT - IV

Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €-Productions. Chomsky Normal form Greibach Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications **Closure Properties of Context-Free Languages:** Closure properties of CFL's, Decision Properties of CFL's Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

UNIT - V

Types of Turing machine: Turing machines and halting

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

Course Outcomes:

CO-1: Explain basic concepts in formal language theory, grammars, automata theory (DFA&NFA), computability theory, and complexity theory.
CO-2: Know the production rules of regular expressions and closure properties of regular languages.
CO3: Construct a pushdown automata and context free grammar
CO4: Understand the concepts of Normal forms and Evaluate solution for various problems using a theoretical computer (Turing machine)
CO5:Distinguish between decidability and undecidability.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd edition, PHI.

REFERENCE BOOKS:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Textbook on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS41) ARTIFICIAL INTELLIGENCE

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Prerequisites:

1. Programming for problem solving, Data Structures.

Course Objectives:

- To learn the distinction between optimal reasoning Vs. human like reasoning
- To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
- To learn different knowledge representation techniques.
- To understand the applications of AI, namely game playing, theorem proving, and machine learning.

UNIT - I

Introduction to AI, Intelligent Agents, problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search, Iterative deepening Depth-first search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions, Beyond Classical Search: Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces

UNIT - II

Problem Solving by Search-II and Propositional Logic

Adversarial Search: Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real-Time Decisions. Constraint Satisfaction Problems: Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems. Propositional Logic: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

UNIT - III

Logic and Knowledge Representation

First-Order Logic: Representation, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

UNIT - IV

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

Classical Planning: Definition of Classical Planning, Algorithms for Planning with State-Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.

UNIT - V

Uncertain knowledge and Learning Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use **Probabilistic Reasoning:** Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.

Course Outcomes:

1. Understand search strategies and intelligent agents
2. Understand different adversarial search techniques
3. Apply propositional logic, predicate logic for knowledge representation
4. Understand the concepts of Classical Planning and reasoning systems
5. Distinguish between certainty and Uncertainty and probabilistic reasoning

TEXT BOOK:

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henry Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS42) FULL STACK DEVELOPMENT
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Pre-Requisites:
1. Object Oriented Programming
2. Web Technologies

Course Objectives:
1. Students will become familiar to implement fast, efficient, interactive and scalable web applications using run time environment provided by the full stack components.

UNIT-I

Introduction to Full Stack Development:
Understanding the Basic Web Development Framework- User, Browser, Webserver, Backend Services, Full Stack Components - Node.js, MongoDB, Express, React, Angular. Java Script Fundamentals, NodeJS- Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks

UNIT-II

Node.js:
Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules-Using the os Module, Using the util Module, Using the dns Module, Using the crypto Module.

UNIT-III

MongoDB:
Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections,

Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections

UNIT-IV

Express and Angular:

Getting Started with Express, Configuring Routes, Using Requests Objects, Using Response Objects. Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

UNIT-V

React:

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Life Cycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

Course Outcomes:

1. Understand Full stack components for developing web application.
2. Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.
3. Understand MongoDB database for storing and processing huge data and connects with NodeJS application.
4. Design faster and effective single page applications using Express and Angular.
5. Create interactive user interfaces with react components.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS43) INTERNET OF THINGS
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Pre-Requisites: Computer organization, Computer Networks

Course Objectives:

- To introduce the terminology, technology and its applications
- To introduce the concept of M2M (machine to machine) with necessary protocols
- To introduce the Python Scripting Language which is used in many IoT devices
- To introduce the Raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of web-based services on IoT devices

UNIT - I

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels and Deployment Templates

Domain Specific IoTs – Home automation, Environment, Agriculture, Health and Lifestyle

UNIT - II

IoT and M2M – M2M, Difference between IoT and M2M, SDN and NFV for IoT,

IoT System Management with NETCOZF, YANG- Need for IoT system Management, Simple Network management protocol, Network operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG

UNIT - III

IoT Systems – Logical design using Python-Introduction to Python – Python Data types & Data structures, Control flow, Functions, Modules, Packaging, File handling, Data/Time operations, Classes, Exception, Python packages of Interest for IoT

UNIT - IV

IoT Physical Devices and Endpoints - Raspberry Pi, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry PI with Python, Other IoT devices.

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Xively Cloud for IoT, Python web application framework
–Django, Designing a RESTful web API

UNIT V

Case studies- Home Automation, Environment-weather monitoring-weather reporting- air pollution monitoring, Agriculture.

Course Outcomes:

1. Interpret the vision of IoT from a global context.
2. Perceive building blocks of Internet of Things and its characteristics.
3. Learn the basic concepts of Python. Implement python programming using Raspberry.
4. Perceive the application areas of IoT. Realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
5. Determine the Market perspective of IoT. Develop Python web applications and cloud servers for IoT.

TEXT BOOK:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547.

REFERENCE BOOK:

1. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING

1. (B22CS44) SCRIPTING LANGUAGES

(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Prerequisites:

1. A course on "Computer Programming and Data Structures".
2. A course on "Object Oriented Programming Concepts".

Course Objectives:

- This course introduces the script programming paradigm
- Introduces scripting languages such as Perl, Ruby and TCL.
- Learning TCL

UNIT - I

Introduction: Ruby, Rails, The structure and Execution of Ruby Programs, Package Management with RUBYGEMS, Ruby and web: Writing CGI scripts, cookies, Choice of Webservers, SOAP and web services

RubyTk – Simple Tk Application, widgets, Binding events, Canvas, scrolling

UNIT - II

Extending Ruby: Ruby Objects in C, the Jukebox extension, Memory allocation, Ruby Type System, Embedding Ruby to Other Languages, Embedding a Ruby Interpreter

UNIT - III

Introduction to PERL and Scripting

Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

UNIT - IV

Advanced perl

Finer points of looping, pack and unpack, filesystem, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

UNIT-V

TCL

TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security Issues, C Interface.

Tk

Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk.

Course Outcomes:

1. Understand Structure and execution of ScriptingLanguages.
2. Identify and apply suitable programming paradigm for a given computing web application.
3. Comprehend the differences between typicalscripting languages and typical system and application programming languages.
4. Gain knowledge of the strengths and weakness of Perl, TCL and Ruby, and select an appropriate language for solving a given problem.
5. Acquire programming skills in scripting language.

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pramatic Progammers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS45) MOBILE APPLICATION DEVELOPMENT
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Prerequisites

1. Acquaintance with JAVA programming
2. A Course on DBMS

Course Objectives

0. To demonstrate their understanding of the fundamentals of Android operating systems
1. To improves their skills of using Android software development tools
2. To demonstrate their ability to develop software with reasonable complexity on mobile platform
3. To demonstrate their ability to deploy software to mobile devices
4. To demonstrate their ability to debug programs running on mobile devices

UNIT - I

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes

Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

UNIT - II

Android User Interface: Measurements – Device and pixel density independent measuring unit - s Layouts – Linear, Relative, Grid and Table Layouts

User Interface (UI) Components –Editable and non-editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers

Event Handling – Handling clicks or changes of various UI components

Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

UNIT - III

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS

Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity

Notifications – Creating and Displaying notifications, Displaying Toasts

UNIT - IV

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference

UNIT - V

Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and etindelg data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

Course Outcomes:

1. Understand the working of Android OS Practically.
2. Develop Android user interfaces
3. Develop, deploy and maintain the Android Applications using Intents and Broadcasts.
4. Understand and use the Persistent Storage Tool
5. Develop and deploy the Android Applications using SQLite database.

TEXT BOOK:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012.

REFERENCE BOOKS:

1. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013
2. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS46) SOFTWARE TESTING METHODOLOGIES
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Prerequisites
1. Software Engineering

Course Objectives
0. To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
1. To develop skills in software test automation and management using the latest tools.

UNIT - I
Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II
Transaction Flow Testing: transaction flows, transaction flow testing techniques.
Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flow testing.
Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - III
Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV
State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V
Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like Jmeter/selenium/soapUI/Catalon).

Course Outcomes:

1. Understand purpose of testing and path testing.
2. Understand strategies in data flow testing and domain testing.
3. Develop logic-based test strategies.
4. Understand graph matrices and its applications.
5. Implement test cases using any testing automation tool.

TEXT BOOKS:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS11) DATA STRUCTURES
(Open Elective – I)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Prerequisites

1. A course on "Programming for Problem Solving

Course Objectives

- Exploring basic data structures such as stacks and queues.
- Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
- Introduces sorting and pattern matching algorithms

UNIT - I

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks-Operations, array and linked representations of stacks, stack applications, Queues-operations, array and linked representations.

UNIT - II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.

Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing- linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

Search Trees: Binary Search Trees, Definition, Implementation, Operations-Searching, Insertion and Deletion, B- Trees, B+ Trees, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching, Red –Black, Splay Trees.

UNIT - IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.
Sorting: Quick Sort, Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer – Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

Course Outcomes:

1. To build the basic knowledge to handle operations like insertions, deletions, searching, and traversing mechanisms in linear data structures.
2. Understand Dictionaries, skip list and Hashing Techniques.
3. Ability to have knowledge on general tree structures, search trees, AVL-trees, B-Trees, B+ Trees, Red-Black Trees, Splay Trees.
4. Apply suitable algorithms for graph traversing and sorting techniques.
5. Implement and know the application of algorithms for pattern matching and Tries.

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2 nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.
2. Data Structures using C – A. S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

REFERENCE BOOK:

1. Data Structures: A Pseudocode Approach with C, 2 nd Edition, R. F. Gilberg and B.A.Forouzan, Cengage Learning.

VAAGDEVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
(B22CS13) OBJECT ORIENTED PROGRAMMING THROUGH JAVA
(Open Elective – I)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Course Objectives

- To Understand the basic object-oriented programming concepts and apply them in problem solving.
- To Illustrate inheritance concepts for reusing the program.
- To Demonstrate multitasking by using multiple threads and event handling
- To Develop data-centric applications using JDBC.
- To Understand the basics of java console and GUI based programming

UNIT - I

Object oriented thinking and Java Basics- Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms. A way of viewing world – Agents, responsibility, messages, methods, History of Java, Java buzzwords, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, parameter passing, recursion, nested and inner classes, exploring string class.

UNIT - II

Inheritance, Packages and Interfaces – Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT - III

Exception handling and Multithreading-- Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. String handling, Exploring java.util. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread

communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

UNIT - IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, checkbox groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

UNIT - V

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Course Outcomes:

1. Understand the use of OOP concepts and solve real world problems using OOP techniques.
2. Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.
3. Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.
4. Able to gain the knowledge on Event handling and develop web related applications using AWT.
5. Design and develop GUI based applications using Applets and Swings.

TEXT BOOKS:

1. Java the complete reference, 7th edition, Herbert schildt, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley & sons.
2. An Introduction to OOP, third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y. Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A. Johnson- Thomson.
5. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.
6. Core Java 2, Vol 2, Advanced Features, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.
7. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.

8. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.
9. Maurach's Beginning Java2 JDK 5, SPD.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS18) DATABASE MANAGEMENT SYSTEMS
(Open Elective – I)

B.Tech III Year II Sem. **L T P C**
3 0 0 3

Prerequisites: A course on "Data Structures".

Course Objectives:
- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

UNIT - I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT - II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views.

Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT - III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and

EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols,

Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

UNIT - V

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM),

B+ Trees: A Dynamic Index Structure.

Course Outcomes:

1. Describe the fundamental elements of relational database management systems
2. Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.
3. Convert the ER-model to relational tables, and improve the database design by normalization Techniques.
4. Apply the principles of database transaction management, database recovery and security.
5. Familiar with basic database storage structures and access techniques file and page organizations, indexing methods including B tree, and hashing.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, McGraw hill, V edition. 3rd Edition
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22AI08) MACHINE LEARNING LAB

B.Tech III Year II Sem. **L T P C**
0 0 2 1

Course Objective:

0. The objective of this lab is to get an overview of the various machine learning techniques and can demonstrate them using python.

List of Experiments

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, Standard Deviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy
3. Study of Python Libraries for ML application such as Pandas and Matplotlib
4. Write a Python program to implement Simple Linear Regression
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn.
6. Implementation of Decision tree using sklearn and its parameter tuning.
7. Implementation of KNN using sklearn.
8. Implementation of Logistic Regression using sklearn.
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

Course Outcomes:

1. Understand modern notions in predictive data analysis.
2. Select data, model selection, model complexity and identify the trends
3. Understand a range of machine learning algorithms along with their strengths and weaknesses.
4. Build predictive models from data and analyze their performance.

TEXT BOOK:

1. Machine Learning – Tom M. Mitchell, - MGH.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & F

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS47) ARTIFICIAL INTELLIGENCE LAB

B.Tech III Year II Sem. **L T P C**
0 0 2 1

Course Objectives:

0. Become familiar with basic principles of AI toward problem solving, knowledge representation, and learning.

LIST OF EXPERIMENTS:

Write a Program to Implement the following using Python.

1. Breadth First Search
2. Depth First Search
3. Tic-Tac-Toe game
4. 8-Puzzle problem
5. Water-Jug problem
6. Travelling Salesman Problem
7. Tower of Hanoi
8. Monkey Banana Problem
9. Alpha-Beta Pruning
10. 8-Queens Problem

Course Outcomes:

1. To Use Control Structures and Operators to write basic Python programming.
2. To Analyze object-oriented concepts in Python.
3. To Evaluate the AI models pre-processed through various feature engineering algorithms by Python Programming.
4. To Develop the code for the recommender system using Natural Language processing.

TEXT BOOK:

1. Artificial Intelligence a Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS48) FULL STACK DEVELOPMENT LAB
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
0 0 2 1

Pre-Requisites:
1. Object Oriented Programming
2. Web Technologies

Course Objectives:
0. Introduce fast, efficient, interactive and scalable web applications using run time environment provided by the full stack components.

List of Experiments
1. Create an application to setup node JS environment and display "Hello World".
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write and other operations on a file.
4. Write a Node JS program to read form data from query string and generate response using NodeJS
5. Create a food delivery website where users can order food from a particular restaurant listed in the website for handling http requests and responses using NodeJS.
6. Implement a program with basic commands on databases and collections using MongoDB.
7. Implement CRUD operations on the given dataset using MongoDB.
8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.
9. Develop an angular JS form to apply CSS and Events.
10. Develop a Job Registration form and validate it using angular JS.
11. Write an angular JS application to access JSON file data of an employee from a server using
    $http service.
12. Develop a web application to manage student information using Express and Angular JS.
13. Write a program to create a simple calculator Application using React JS.
14. Write a program to create a voting application using React JS
15. Develop a leave management system for an organization where users can apply different types of leaves such as casual leave and medical leave. They also can view the available number of days using react application.
16. Build a music store application using react components and provide routing among the web pages.
17. Create a react application for an online store which consist of registration, login, product information pages and implement routing to navigate through these pages.

Course Outcomes:

1. Design flexible and responsive Web applications using Node JS, React, Express and Angular.
2. Perform CRUD operations with MongoDB on huge amount of data.
3. Develop real time applications using react components.
4. Use various full stack modules to handle http requests and responses.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley,2019.
2. Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress,2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. Brad Green& Seshadri. Angular JS. 1st Edition. O'Reilly Media, 2013.
4. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS49) INTERNET OF THINGS LAB
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
0 0 2 1

Course Objectives

0. To introduce the raspberry PI platform, that is widely used in IoT applications
1. To introduce the implementation of distance sensor on IoT devices

List of Experiments

1. Using Raspberry pi
   a. Calculate the distance using a distance sensor.
   b. Interface an LED and switch with Raspberry pi.
   c. Interface an LDR with Raspberrry Pi.

2. Using Arduino
   a. Calculate the distance using a distance sensor.
   b. Interface an LED and switch with Aurdino.
   c. Interface an LDR with Aurdino
   d. Calculate temperature using a temperature sensor.

3. Using Node MCU
   a. Calculate the distance using a distance sensor.
   b. Interface an LED and switch with Raspberry pi.
   c. Interface an LDR with Node MCU
   d. Calculate temperature using a temperature sensor.

4. Installing OS on Raspberry Pi
   a) Installation using PiImager
   b) Installation using image file
      - Downloading an Image
      - Writing the image to an SD card
        - using Linux
        - using Windows
      - Booting up Follow the instructions given in the URL https://www.raspberrypi.com/documentation/computers/getting-started.html

5. Accessing GPIO pins using Python
   a) Installing GPIO Zero

library. update your
repositories list:
install the package for Python 3:

b) Blinking an LED connected to one of the GPIO pin

c) Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) using the in-built PWM wavelength.

6. Create a DJANGO project and an app.
7. Create a DJANGO view for weather station REST API
8. Create DJANGO template
9. Configure MYSQL with DJANGO framework

Course Outcomes

1. Ability to introduce the concept of M2M (machine to machine) with necessary protocols and get awareness in implementation of distance sensor
2. Get the skill to program using python scripting language which is used in many IoT devices
3. Illustrate different sensor technologies for sensing real world entities and identify the applications of IoT in Industry
4. Appraise the role of IoT protocols for efficient network communication

TEXT BOOKS:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547.
2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

REFERENCE BOOKS:

1. Bernd Scholz-Reiter, Florian Michahelles, “Architecting the Internet of Things”, ISBN 978-3- 642-19156-5 e-ISBN 978-3-642-19157-2, Springer, 2016
2. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS50) SCRIPTING LANGUAGES LAB
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
0 0 2 1

Prerequisites: Any High level programming language (C, C++)

Course Objectives

0. To Understand the concepts of scripting languages for developing web based projects
1. To understand the applications the of Ruby, TCL, Perl scripting languages

LIST OF EXPERIMENTS

1. Write a Ruby script to create a new string which is n copies of a given string where n is a non-negative integer
2. Write a Ruby script which accept the radius of a circle from the user and compute the parameter and area.
3. Write a Ruby script which accept the users first and last name and print them in reverse order with a space between them
4. Write a Ruby script to accept a filename from the user print the extension of that
5. Write a Ruby script to find the greatest of three numbers
6. Write a Ruby script to print odd numbers from 10 to 1
7. Write a Ruby script to check two integers and return true if one of them is 20 otherwise return their sum
8. Write a Ruby script to check two temperatures and return true if one is less than 0 and the other is greater than 100
9. Write a Ruby script to print the elements of a given array
10. Write a Ruby program to retrieve the total marks where subject name and marks of a student stored in a hash
11. Write a TCL script to find the factorial of a number
12. Write a TCL script that multiplies the numbers from 1 to 10
13. Write a TCL script for sorting a list using a comparison function
14. Write a TCL script to (i) create a list (ii) append elements to the list (iii) Traverse the list (iv) Concatenate the list
15. Write a TCL script to comparing the file modified times.
16. Write a TCL script to Copy a file and translate to native format.
17. a) Write a Perl script to find the largest number among three numbers.
    b) Write a Perl script to print the multiplication tables from 1-10 using subroutines.
18. Write a Perl program to implement the following list of manipulating functions
    a) Shift
    b) Unshift

c) Push

19. a) Write a Perl script to substitute a word, with another word in a string.
    b) Write a Perl script to validate IP address and email address.
20. Write a Perl script to print the file in reverse order using command line arguments

Course Outcomes:

1. Able to use Scripting languages compiler and platform to write and execute scripting languages program.
2. Understand and Apply Object oriented features and Scripting languages concepts.
3. Ability to understand the differences between Scripting languages and programming languages.
4. Able to gain some fluency programming in Ruby, Perl, TCL.

TEXT BOOKS:

1. The World of Scripting Languages, David Barron,Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pramatic Progammers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS51) MOBILE APPLICATION DEVELOPMENT LAB
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
0 0 2 1

Prerequisites: --- NIL---

Course Objectives:

- To learn how to develop Applications in an android environment.
- To learn how to develop user interface applications.
- To learn how to develop URL related applications.

LIST OF EXPERIMENTS:

1. Create an Android application that shows Hello + name of the user and run it on an emulator.

(b) Create an application that takes the name from a text box and shows hello message along with the name entered in the text box, when the user clicks the OK button.

2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Datepicker), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on the right fragment instead of the second screen with the back button. Use Fragment transactions and Rotation event listeners.
4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.
5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6. Create an application that uses a text file to store usernames and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with a Login Failed message.
7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords

are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.

9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.
10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.
11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.
12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

Course Outcomes:

1. Student will be able to develop, deploy and maintain the Android Applications.
2. Understand the working of Android OS Practically.
3. Ability to develop the user interface.
4. Ability to work with SQLITE DB.

TEXT BOOKS:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012.
2. Android Application Development for Java Programmers, James C Sheusi, Cengage, 2013.

REFERENCE BOOK:

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS52) SOFTWARE TESTING METHODOLOGIES LAB
(Professional Elective – III)

B.Tech III Year II Sem. **L T P C**
0 0 2 1

Prerequisites

- A basic knowledge of programming.

Course Objectives

- To provide knowledge of software testing methods.
- To develop skills in automation of software testing and software test automation management using the latest tools.

List of Experiments:

1. Recording in context sensitive mode and analog mode
2. GUI checkpoint for single property
3. GUI checkpoint for single object/window
4. GUI checkpoint for multiple objects 5.
   a. Bitmap checkpoint for object/window
   b. Bitmap checkpoint for screen area
6. Database checkpoint for Default check
6. Database checkpoint for custom check
6. Database checkpoint for runtime record check 6.
   a. Data driven test for dynamic test data submission
   b. Data driven test through flat files
   c. Data driven test through front grids
   d. Data driven test through excel test

   a. Batch testing without parameter passing
   b. Batch testing with parameter passing
11. Data driven batch
11. Silent mode test execution without any interruption
12. Test case for calculator in windows application

Course Outcomes:

1. Identify Recording in context sensitive mode and analog mode. (Knowledge)
2. Analyze the GUI checkpoint for single property, object/window, multiple objects. (Analysis
3. Identify Bitmap checkpoint for object/window, screen area(Knowledge)

4. Summarize Database checkpoint for Default check, custom check, runtime record check(Evaluation)
5. Identify Data driven test through flat files, excel test(Knowledge) C328.6 Recognize Batch testing without parameter passing, with parameter passing. (Knowledge)

TEXT BOOKS:

1. Software Testing techniques, Baris Beizer, 2nd Edition, Dreamtech.
2. Software Testing Tools, Dr. K.V.K.K.Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing, Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World, Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing, Meyers, John Wiley.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS53) BIG DATA-SPARK

B.Tech III Year II Sem. **L T P C**
0 0 4 2

Course Objectives:

- The main objective of the course is to process Big Data with advance architecture like spark and streaming data in Spark

List of Experiments:

1. To Study of Big Data Analytics and Hadoop Architecture
   (i) know the concept of big data architecture
   (ii) know the concept of Hadoop architecture

2. Loading DataSet in to HDFS for
   Spark Analysis Installation of
   Hadoop and cluster management
   (i) Installing Hadoop single node cluster in ubuntu environment
   (ii) Knowing the differencing between single node clusters and multi-node clusters
   (iii) Accessing WEB-UI and the port number
   (iv) Installing and accessing the environments such as hive and sqoop

3. File management tasks & Basic linux commands
   (i) Creating a directory in HDFS
   (ii) Moving forth and back to directories
   (iii) Listing directory contents
   (iv) Uploading and downloading a file in HDFS
   (v) Checking the contents of the file
   (vi) Copying and moving files
   (vii) Copying and moving files between local to HDFS environment
   (viii) Removing files and paths
   (ix) Displaying few lines of a file
   (x) Display the aggregate length of a file
   (xi) Checking the permissions of a file
   (xii) Zipping and unzipping the files with & without permission pasting it to a location
   (xiii) Copy, Paste commands

4. Map-reducing
   (i) Definition of Map-reduce
   (ii) Its stages and terminologies
   (iii) Word-count program to understand map-reduce (Mapper phase, Reducer phase,

Driver code)

5. Implementing Matrix-Multiplication with Hadoop Map-reduce

6. Compute Average Salary and Total Salary by Gender for an Enterprise.

7. (i) Creating hive tables (External and internal)
   (ii) Loading data to external hive tables from sql tables(or)Structured c.s.v using scoop
   (iii) Performing operations like filterations and updations
   (iv) Performing Join (inner, outer etc)
   (v) Writing User defined function on hive tables

8. Create a sql table of employees Employee table with id,designation Salary table (salary ,dept id) Create external table in hive with similar schema of above tables,Move data to hive using scoop and load the contents into tables,filter a new table and write a UDF to encrypt the table with AES-algorithm, Decrypt it with key to show contents

9. (i) Pyspark Definition(Apache Pyspark) and difference between Pyspark, Scala, pandas
   (ii) Pyspark files and class methods
   (iii) get(file name)
   (iv) get root directory()

10. Pyspark -RDD'S
    (i) what is RDD's?
    (ii) ways to Create RDD
    (iii) parallelized collections
    (iv) external dataset
    (v) existing RDD's
    (vi) Spark RDD's operations (Count, foreach(), Collect, join,Cache()

11. Perform pyspark transformations
    (i) map and flatMap
    (ii) to remove the words, which are not necessary to analyze this text.
    (iii) groupBy
    (iv) What if we want to calculate how many times each word is coming in corpus ?
    (v) How do I perform a task (say count the words 'spark' and 'apache' in rdd3) separatly on each partition and get the output of the task performed in these partition ?
    (vi) unions of RDD
    (vii) join two pairs of RDD Based upon their key

12. Pyspark sparkconf-Attributes and applications
    (i) What is Pyspark spark conf ()
    (ii) Using spark conf create a spark session to write a dataframe to read details in a c.s.v and later move that c.s.v to another location.

Course Outcomes:

1. Develop MapReduce Programs to analyze large dataset Using Hadoop and Spark.
2. Write Hive queries to analyze large dataset Outline the Spark Ecosystem and its components.
3. Perform the filter, count, distinct, map, flatMap RDD Operations in Spark.
4. Build Queries using Spark SQL.
5. Apply Spark joins on Sample Data Sets and make use of sqoop to import and export data from hadoop to database and vice-versa.

TEXT BOOKS:

1. Spark in Action, Marko Bonaci and Petar Zecevic, Manning.
2. PySpark SQL Recipes: With HiveQL, Dataframe and Graphframes, Raju Kumar Mishra and Sundar Rajan Raman, Apress Media.

WEB LINKS:

1. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01330150 5844518912251 8 2_shared/overview
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012583881 19638835242_s hared/overview
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01260526 8423008256169 2 _shared/overview

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CH03) ENVIRONMENTAL SCIENCE

B.Tech III Year II Sem. **L T P C**
3 0 0 0

Course Objectives:

0. Understanding the importance of ecological balance for sustainable development.
1. Understanding the impacts of developmental activities and mitigation measures.
2. Understanding the environmental policies and regulations

UNIT - I

Ecosystems: Definition, Scope, and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Biomagnification, ecosystem value, services and carrying capacity, Field visits.

UNIT - II

Natural Resources: Classification of Resources: Living and Non-Living resources, **water resources:** use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. **Mineral resources:** use and exploitation, environmental effects of extracting and using mineral resources, **Land resources:** Forest resources, **Energy resources:** growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

UNIT - III

Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation. National Biodiversity act.

UNIT - IV

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, **Air Pollution:** Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. **Water pollution:** Sources and types of pollution, drinking water quality standards. **Soil Pollution:** Sources and types, Impacts of modern agriculture, degradation of soil. **Noise Pollution:** Sources and Health hazards, standards, **Solid waste:** Municipal Solid Waste management, composition and characteristics of e-Waste and its management. **Pollution control technologies:** Wastewater Treatment methods: Primary, secondary and Tertiary.
Overview of air pollution control technologies, Concepts of bioremediation. **Global**

Environmental Issues and Global Efforts: Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol. NAPCC-GoI Initiatives.

UNIT - V

Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio-economical aspects. Strategies for risk assessment, Concepts of Environmental Management Plan

(EMP). **Towards Sustainable Future:** Concept of Sustainable Development Goals, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

Course Outcomes:

1. Based on this course, the Engineering graduate will understand /evaluate / develop technologies on the basis of ecological principles and environmental regulations which in turn helps in sustainable development

TEXT BOOKS:

1 Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
2 Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.
2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.
3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
5. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.
6. Introduction to Environmental Science by Y. Anjaneyulu, BS. Publications.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING

(B22CS53) Industrial Oriented Mini Project/ Internship/ Skill Development Course
(Big data-Spark)

B.Tech III Year II Sem.

L	**T**	**P**	**C**
0	0	4	2

Course Outcomes:

1. Enhance students' knowledge in current technology.
2. Develop leadership ability and responsibility to execute the given task.
3. Choose efficient tools for designing project modules.
4. Enhance their employability skills along with real corporate exposure.
5. Elaborate the completed task and compile the report.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS54) CRYPTOGRAPHY AND NETWORK SECURITY

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

- Explain the importance and application of each of confidentiality, integrity, authentication and availability
- Understand various cryptographic algorithms.
- Understand the basic categories of threats to computers and networks
- Describe public-key cryptosystem.
- Describe the enhancements made to IPv4 by IPSec
- Understand Intrusions and intrusion detection

UNIT - I

Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security **Cryptography Concepts and Techniques:** Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

UNIT - II

Symmetric key Ciphers: Block Cipher principles, DES, AES, Blowfish, RC5, IDEA, Block cipher operation, Stream ciphers, RC4.

Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange, Knapsack Algorithm.

UNIT - III

Cryptographic Hash Functions: Message Authentication, Secure Hash Algorithm (SHA-512), **Message authentication codes:** Authentication requirements, HMAC, CMAC, Digital signatures, Elgamal Digital Signature Scheme.

Key Management and Distribution: Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure

UNIT - IV

Transport-level Security: Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH)

Wireless Network Security: Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security

UNIT - V

E-Mail Security: Pretty Good Privacy, S/MIME IP Security: IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet Key Exchange

Case Studies on Cryptography and security: Secure Multiparty Calculation, Virtual Elections, Single sign On, Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability

Course Outcomes:

1. Student will be able to understand basic cryptographic concepts and techniques.
2. Compare and contrast symmetric and Asymmetric Algorithms.
3. Implementation of message authentication,Hashing Algorithms and able to understand Kerberos.
4. Ability to understand the current legal issues towards information security.
5. Explore the attacks and control associated with IP Security,Email Security

TEXT BOOKS:

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition
2. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition

REFERENCE BOOKS:

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition.
2. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition
3. Information Security, Principles, and Practice: Mark Stamp, Wiley India.
4. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH
5. Introduction to Network Security: Neal Krawetz, CENGAGE Learning
6. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS55) COMPILER DESIGN

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Prerequisites

1. A course on "Formal Languages and Automata Theory".
2. A course on "Computer Organization and architecture".
3. A course on "Data Structures".

Course Objectives:

0. Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
1. Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, code optimization techniques, intermediate code generation, code generation and data flow analysis.

UNIT - I

Introduction: The structure of a compiler, the science of building a compiler, programming language basics

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyzer Generator, Optimization of DFA-Based Pattern Matchers.

UNIT - II

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers, Using Ambiguous Grammars and Parser Generators.

UNIT - III

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax-Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's.

Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code, Types and Declarations, Type Checking, Control Flow, Switch-Statements, Intermediate Code for Procedures.

UNIT - IV

Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management, Introduction to Garbage Collection, Introduction to

Trace-Based Collection.

Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, A Simple Code Generator, Peephole Optimization, Register Allocation and Assignment, Dynamic Programming Code-Generation

UNIT - V

Machine-Independent Optimization: The Principal Sources of Optimization, Introduction to Data-Flow Analysis, Foundations of Data-Flow Analysis, Constant Propagation, Partial-Redundancy Elimination, Loops in Flow Graphs.

Course Outcomes:

1. Demonstrate the ability to design a compiler given a set of language features.
2. Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
3. Acquire skills in using lex tool & yacc tool for developing a scanner and parser.
4. Design and implement LL and LR parsers
5. Design algorithms to perform code optimization and generate machine code in order to improve the performance of a program in terms of space and time complexity.

TEXT BOOK:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman.

REFERENCE BOOKS:

1. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Louden, Thomson.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS56) GRAPH THEORY
(Professional Elective – IV)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

- Understanding graphs, trees, connected paths, applications of trees and graphs.

UNIT - I

Introduction-Discovery of graphs, Definitions, Subgraphs, Isomorphic graphs, Matrix representations of graphs, Degree of a vertex, Directed walks, paths and cycles, Connectivity in digraphs, Eulerian and Hamilton digraphs, Eulerian digraphs, Hamilton digraphs, Special graphs, Complements, Larger graphs from smaller graphs, Union, Sum, Cartesian Product, Composition, Graphic sequences, Graph theoretic model of the LAN problem, Havel-Hakimi criterion, Realization of a graphic sequence.

UNIT - II

Connected graphs and shortest paths - Walks, trails, paths, cycles, Connected graphs, Distance, Cut-vertices and cut-edges, Blocks, Connectivity, Weighted graphs and shortest paths, Weighted graphs, Dijkstra‟s shortest path algorithm, Floyd-Warshall shortest path algorithm.

UNIT - III

Trees- Definitions and characterizations, Number of trees, Cayley‟s formula, Kircho↵-matrix-tree theorem, Minimum spanning trees, Kruskal‟s algorithm, Prim‟s algorithm, Special classes of graphs, Bipartite Graphs, Line Graphs, Chordal Graphs, Eulerian Graphs, Fleury‟s algorithm, Chinese Postman problem, Hamilton Graphs, Introduction, Necessary conditions and sufficient conditions.

UNIT - IV

Independent sets coverings and matchings– Introduction, Independent sets and coverings: basic equations, Matchings in bipartite graphs, Hall‟s Theorem, K¨onig‟s Theorem, Perfect matchings in graphs, Greedy and approximation algorithms.

UNIT - V

Vertex Colorings- Basic definitions**,** Cliques and chromatic number, Mycielski‟s theorem, Greedy coloring algorithm, Coloring of chordal graphs, Brooks theorem, Edge Colorings, Introduction and Basics, Gupta-Vizing theorem, Class-1 and Class-2 graphs, Edge-coloring of bipartite graphs, Class-2 graphs, Hajos union and Class-2 graphs, A scheduling problem and equitable edge-coloring.

Course Outcomes:

1. Ability to understand some important classes of graph theoretic models.
2. Prove central theorems about trees, matching, connectivity, coloring and planar graphs.
3. Describe and implement basic algorithms for graphs.
4. Use graph theory as a modeling tool.
5. Learn different coloring algorithms for graphs.

TEXT BOOKS:

1. J. A. Bondy and U. S. R. Murty. Graph Theory, volume 244 of Graduate Texts in Mathematics. Springer, 1st edition, 2008.
2. J. A. Bondy and U. S. R. Murty. Graph Theory with Applications.

REFERENCE BOOKS:

1. Lecture Videos: http://nptel.ac.in/courses/111106050/13
2. Introduction to Graph Theory, Douglas B. West, Pearson.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS57) CYBER SECURITY
(Professional Elective – IV)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course objectives:

- To understand various types of cyber-attacks and cyber-crimes.
- To learn threats and risks within the context of cyber security.
- To have an overview of the cyber laws & concepts of cyber forensics.
- To study the defensive techniques against these attacks.

UNIT- I

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

UNIT - II

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy.

Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics.

UNIT - III

Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops.

UNIT- IV

Cyber Security: Organizational Implications: Introduction, cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing and the associated challenges for organizations.

UNIT - V

Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Data linking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc.

Cybercrime: Examples and Mini-Cases

Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances.

Mini-Cases: The Indian Case of online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

Course Outcomes:

1. Analyze and evaluate the cyber security needs of an organization.
2. Understand Cyber Security Regulations and Roles of International Law.
3. To learn the vulnerabilities of mobile and wireless devices.
4. Design and develop security architecture for an organization.
5. Understand fundamental concepts of data privacy attacks.

TEXT BOOKS:

1. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley

REFERENCE BOOKS:

1. B. B. Gupta, D.P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press
2. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
3. Introduction to Cyber Security, Chwan-Hwa(john) Wu, J.David Irwin, CRC Press T&F Group.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS58) SOFT COMPUTING
(Professional Elective – IV)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

- Familiarize with soft computing concepts
- Introduce and use the idea of fuzzy logic and use of heuristics based on human experience
- Familiarize the Neuro-Fuzzy modeling using Classification and Clustering techniques
- Learn the concepts of Genetic algorithm and its applications
- Acquire the knowledge of Rough Sets.

UNIT - I

Introduction to Soft Computing: Evolutionary Computing, "Soft" computing versus "Hard" computing, Soft Computing Methods, Recent Trends in Soft Computing, Characteristics of Soft computing, Applications of Soft Computing Techniques.

UNIT- II

Fuzzy Systems: Fuzzy Sets, Fuzzy Relations, Fuzzy Logic, Fuzzy Rule-Based Systems

UNIT- III

Fuzzy Decision Making, Particle Swarm Optimization

UNIT- IV

Genetic Algorithms: Basic Concepts, Basic Operators for Genetic Algorithms, Crossover and Mutation Properties, Genetic Algorithm Cycle, Fitness Function, Applications of Genetic Algorithm.

UNIT- V

Rough Sets, Rough Sets, Rule Induction, and Discernibility Matrix, Integration of Soft Computing Techniques.

Course Outcomes:

1. Identify the difference between Conventional Artificial Intelligence to Computational Intelligence.
2. Understand fuzzy logic and reasoning to handle and solve engineering problems
3. Apply the Classification techniques on various applications.
4. Perform various operations of genetic algorithms and Rough Sets.

TEXT BOOK:

1. Soft Computing – Advances and Applications - Jan 2015 by B.K. Tripathy and J. Anuradha – Cengage Learning

REFERENCE BOOKS:

1. S. N. Sivanandam & S. N. Deepa, “Principles of Soft Computing”, 2nd edition, Wiley India, 2008.
2. David E. Goldberg, “Genetic Algorithms-In Search, optimization and Machine learning”, Pearson Education.
3. J. S. R. Jang, C.T. Sun and E.Mizutani, “Neuro-Fuzzy and Soft Computing”, Pearson Education, 2004.
4. G.J. Klir & B. Yuan, “Fuzzy Sets & Fuzzy Logic”, PHI, 1995.
5. Melanie Mitchell, “An Introduction to Genetic Algorithm”, PHI, 1998.
6. Timothy J. Ross, “Fuzzy Logic with Engineering Applications”, McGraw-Hill International editions, 1995

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22AI23) CLOUD COMPUTING
(Professional Elective – IV)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Pre-requisites:

1. A course on "Computer Networks".
2. A course on "Operating System".

Course Objectives:

- This course provides an insight into cloud computing
- Topics covered include- Cloud Computing Architecture, Deployment Models, Service Models, Technological Drivers for Cloud Computing, Networking for Cloud Computing and Security in Cloud Computing

UNIT - I

Computing Paradigms, Cloud Computing Fundamentals, Cloud Computing Architecture and Management

UNIT - II

Cloud Deployment Models, Cloud Service Models, Technological Drivers for Cloud Computing: SOA and Cloud, Multicore Technology, Web 2.0 and Web 3.0, Pervasive Computing, Operating System, Application Environment

UNIT - III

Virtualization, Programming Models for Cloud Computing: MapReduce, Cloud Haskell, Software Development in Cloud

UNIT - IV

Networking for Cloud Computing: Introduction, Overview of Data Center Environment, Networking Issues in Data Centers, Transport Layer Issues in DCNs, Cloud Service Providers

UNIT - V

Security in Cloud Computing, and Advanced Concepts in Cloud Computing

Course Outcomes:

1. Understand different computing paradigms and potential of the paradigms and specifically cloud computing
2. Understand cloud service types, cloud deployment models and technologies supporting and driving the cloud
3. Acquire the knowledge of programming models for cloud and development of

software application that runs the cloud and various services available from major cloud providers
4. Understand the security concerns and issues in cloud computing
5. Acquire the knowledge of advances in cloud computing.

TEXT BOOK:

1. Chandrasekaran, K. *Essentials of cloud computing*. CRC Press, 2014

REFERENCE BOOKS:

1. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011
2. Enterprise Cloud Computing - Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010
3. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS59)AD-HOC & SENSOR NETWORKS
(Professional Elective – IV)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Prerequisites

1. Computer Networks
2. Distributed Systems
3. Mobile Computing

Course Objectives

0. To understand the challenges of routing in ad-hoc and sensor networks
1. To understand various broadcast, mutlicast and geocasting protocols in ad hoc and sensor networks
2. To understand basics of Wireless sensors, and Lower Layer Issues and Upper Layer Issues of WSN

UNIT - I

Introduction to Ad Hoc Networks

Characteristics of MANETs, Applications of MANETs and Challenges of MANETs.

Routing in MANETs

Criteria for classification, Taxonomy of MANET routing algorithms, *Topology-based* routing algorithms- Proactive: DSDV, WRP; Reactive: DSR, AODV, TORA; Hybrid: ZRP; *Position- based* routing algorithms- Location Services-DREAM, Quorum-based, GLS; Forwarding Strategies, Greedy Packet, Restricted Directional Flooding-DREAM, LAR; Other routing algorithms-QoS Routing, CEDAR.

UNIT - II

Data Transmission

Broadcast Storm Problem, Rebroadcasting Schemes-Simple-flooding, Probability-based Methods, Area- based Methods, Neighbour Knowledge-based: SBA, Multipoint Relaying, AHBP. Multicasting: Tree-based: AMRIS, MAODV; Mesh-based: ODMRP, CAMP; Hybrid: AMRoute, MCEDAR.

UNIT - III

Geocasting

Data-transmission Oriented-LBM; Route Creation Oriented-GeoTORA, MGR**.**

TCP over Ad Hoc TCP protocol overview, TCP and MANETs, Solutions for TCP over Ad hoc

UNIT - IV

Basics of Wireless Sensors and Lower Layer Issues-Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.

UNIT - V

Upper Layer Issues of WSN

Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

Course Outcomes:

1. Understand the concepts of Routing mechanism in Manets.
2. Understand and compare the MAC and routing protocols for adhoc networks.
3. Understand the geocasting protocols and solutions for Tcp ,adhoc networks.
4. Concepts of classification of sensor networks.
5. Design of layer issues of WSN.

TEXT BOOKS

1. Ad Hoc and Sensor Networks – Theory and Applications, *Carlos Corderio Dharma P.Aggarwal,* World Scientific Publications, March 2006, ISBN – 981-256-681-3
2. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science, ISBN – 978-1-55860-914-3 (Morgan Kauffman)

REFERENCE BOOKS:

1. C. Siva Ram Murthy, B.S. Manoj Ad Hoc Wireless Networks: Architectures and Protocols.
2. Taieb Znati Kazem Sohraby, Daniel Minoli, Wireless Sensor Networks: Technology, Protocols and Applications, Wiley.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS60) ADVANCED ALGORITHMS
(Professional Elective – V)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Pre-Requisites: Algorithm Design and Analysis

Course Objectives:

- To familiarize advanced methods on analysis of algorithms.
- To familiarize with graphs and algorithms related shortest path
- To understand matrix computations and modulo representations
- To introduce randomized, approximation algorithms and computational complexity topics

UNIT – I

Introduction to Algorithms, Classification of Algorithms, Asymptotic Analysis, Introduction to Recurrence equations - Linear recurrences, Non-linear recurrences, Formulation of recurrence equations, techniques for solving recurrence equations, Solving recurrence equations using polynomial reduction, Master's theorem

Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstra's), depth-first search and computation of strongly connected components, Multistage Graph, topological sorting

UNIT – II

Graph Matching: Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path, Bipartite matching problem

Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set, Optimal tree problems- optimal merge, huffman coding, tree vertex splitting problem. **Shortest Path in Graphs:** Floyd-Warshall algorithm, Travelling Sales Person Problem and introduction to dynamic programming paradigm. Optimal Graph Problems - Minimum Spanning Tree, Single source shortest path.

UNIT - III

Flow-Networks: Maxflow - mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.

Matrix Computations: Strassen's algorithm and introduction to divide and conquer paradigm, Chain Matrix Multiplication, Matrix operations – Gaussian Elimination

method, LUP-decomposition, Crout's method of decomposition, inverse of a triangular matrix,

UNIT - IV

Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo-representation, interpolation problem. Multiplication of long integers by using Divide and Conquer paradigm, Schonhage-Strassen's Integer Multiplication algorithm.

String Algorithms: Naïve String, Rabin Karp, KMP, Boyer Moore, Harspool algorithms

UNIT - V

Basics of Computational Complexity: Introduction to computational complexity, complexity classes, Satisfiability problem and Cook's theorem, Examples of NP- Complete problems

Randomized algorithms: Introduction, Types of Randomized algorithms, Example of Randomized algorithms.

Approximation algorithms: Introduction, Types of Approximation algorithms, Examples of Approximation algorithms

Course Outcomes:

1. Ability to Understand the analysis of an algorithm and graph.
2. Understand Matroids and Graph Matching.
3. Ability to Understand Flow-Networks and matrix computations.
4. Understand modulo representations and String Algorithms
5. Categorize the randomized, approximation algorithms and analyse computational complexity.

TEXT BOOK:

1. Design and Analysis of Algorithms, S. Sridhar, Oxford University Press.

REFERENCE BOOKS:

1. Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein.
2. The Design and Analysis of Computer Algorithms, Aho, Hopcroft, Ullman.
3. Algorithm Design, Kleinberg and Tardos.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS61) AGILE METHODOLOGY
(Professional Elective – V)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

- Knowledge on concepts of agile development, releasing, planning and developing.

UNIT - I

Introduction Extreme Programming (XP) - Agile Development

Why Agile?, Understanding Success, Beyond Deadlines, Importance of Organizational Success, Introduction to Agility, Agile methods-Scrum and XP, Manifesto for Agile Software Development, Principles of Agile Process. Understanding XP (Extreme Programming) - XP life cycle, XP team, XP Concepts, Adopting XP - Knowing whether XP is suitable, Implementing XP, assessing Agility, Practicing XP - Thinking, Pair Programming, Energized work, Informative Workspace, Root cause Analysis, Retrospectives.

UNIT - II

Collaborating

Trust, Sit together, Real customer involvement, Ubiquitous language, Stand-Up meetings, coding standards, Iteration demo, Reporting.

UNIT - III

Releasing

Bugfree Release, Version Control, Ten-Minute Build, continuous integration, Collective ownership and Documentation.

UNIT – IV

Planning

Version, Release Planning, The Planning Game, Risk Management, Iteration Planning, Slack, Stories, and Estimating

UNIT - V

Developing

Incremental requirements, Customer tests, Test driven development, Refactoring, Incremental design and architecture, spike solutions, Performance optimization, Exploratory testing.

Course Outcomes:

1. Identify basic concepts of agile methodology and Extreme programming
2. Analyze real customer involvement in collaboration
3. Understand the importance of bugfree release by continuous integration.
4. Discuss risk management and iteration planning
5. Understanding incremental requirements, refactoring, incremental design and architecture

TEXT BOOK:

1. The art of Agile Development, James Shore and Shane Warden, 11th Indian Reprint, O'Reilly, 2018.

REFERENCE BOOKS:

1. Learning Agile, Andrew Stellman and Jennifer Greene, O'Reilly, 4th Indian Reprint, 2018
2. Practices of an Agile Developer, Venkat Subramaniam and Andy Hunt, SPD, 5th Indian Reprint, 2015
3. Agile Project Management - Jim Highsmith, Pearson Low price Edition 2004

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS62) ROBOTIC PROCESS AUTOMATION
(Professional Elective – V)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

- Introduce robotic process automation, techniques of automation using UIPath RPA tool.

UNIT - I

Robotic Process Automation: Introduction**,** Scope and techniques of automation, Robotic process automation, Components of RPA, RPA platforms, About UiPath

UIPath Stack Uipath Studio, Uipath Robot, Types of Robots, UiPath Orchestrator
UIPath Studio Projects, User interface
The User Interface: Task recorder, Advanced UI interactions: Input methods, Output methods

UNIT - II

Sequence, Flowchart, and Control Flow: Sequencing the workflow, Activities, Control Flow, various types of loops and decision making

Data Manipulation: Variables and scope, Collections, Arguments – Purpose and use, Data table usage with examples, File operation with step-by-step example, CSV/Excel to data table and vice versa

UNIT - III

Taking Control of the Controls: Finding and attaching windows, Finding the control, Techniques for waiting for a control, Act on controls – mouse and keyboard activities, Handling events, revisit recorder, When to use OCR, Types of OCR available, How to use OCR

Plugins and Extensions: Terminal Plugin, SAP Automation, Citrix automation and Credential management

UNIT - IV

Handling User Events and Assistant Bots: Assistant bots, Monitoring system event triggers, Monitoring image and element triggers, Launching an assistant bot on a keyboard event

Exception Handling, Debugging, and Logging: Exception handling, Common exceptions and ways to handle them, Logging and taking screenshots, Debugging techniques, Collecting crash dumps, Error reporting

UNIT - V

Managing and Maintaining the Code: Project organization, nesting workflows, Reusability of workflows, Commenting techniques, State Machine, When to use Flowcharts, State Machines, or Sequences, Using config files

Deploying and Maintaining the Bot: Publishing using publish utility, using Orchestration Server to control bots, deploy bots, License Management, Publishing and Managing updates

Course Outcomes:

1. Able to learn the concepts of Robotic Process Automation and Uipath Studio.
2. Flowcharts and diagrams to create their own simple workflow.
3. Gain expertise in Desktop, Web and Citrix Automation.
4. Applying UIPath tool for debugging process
5. Design system managing techniques and Select the packages and to release them for the update of the bots.

TEXT BOOK:

1. Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool - UiPath: Create Software robots. with the leading RPA tool – UiPath Kindle Edition

REFERENCE BOOK:

1. Robotic Process Automation A Complete Guide - 2020 Edition Kindle Edition.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS63) BLOCKCHAIN TECHNOLOGY
(Professional Elective – V)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Prerequisites:

1. Knowledge in information security and applied cryptography.
2. Knowledge in Computer Networks

Course Objectives:

- To learn the fundamentals of Blockchain and various types of block chain and consensus mechanisms.
- To understand the public block chain system, Private block chain system and consortium blockchain.
- Able to know the security issues of blockchain technology.

UNIT-I

Fundamentals of Blockchain: Introduction, Origin of Blockchain, Blockchain Solution, Components of Blockchain, Block in a Blockchain, The Technology and the Future.

Blockchain Types and Consensus Mechanism: Introduction, Decentralization and Distribution, Types of Blockchain, Consensus Protocol.

Cryptocurrency – Bitcoin, Altcoin and Token: Introduction, Bitcoin and the Cryptocurrency, Cryptocurrency Basics, Types of Cryptocurrencies, Cryptocurrency Usage.

UNIT-II

Public Blockchain System: Introduction, Public Blockchain, Popular Public Blockchains, The Bitcoin Blockchain, Ethereum Blockchain.

Smart Contracts: Introduction, Smart Contract, Characteristics of a Smart Contract, Types of Smart Contracts, Types of Oracles, Smart Contracts in Ethereum, Smart Contracts in Industry.

UNIT-III

Private Blockchain System: Introduction, Key Characteristics of Private Blockchain, Need of Private Blockchain, Private Blockchain Examples, Private Blockchain and Open Source, E- commerce Site Example, Various Commands (Instructions) in E-commerce Blockchain, Smart Contract in Private Environment, State Machine, Different Algorithms of Permissioned Blockchain, ByzantineFault, Multichain.

Consortium Blockchain: Introduction, Key Characteristics of Consortium Blockchain, Need of Consortium Blockchain, Hyperledger Platform, Overview of

Ripple, Overview of Corda.
Initial Coin Offering: Introduction, Blockchain Fundraising Methods, Launching an ICO, Investing in an ICO, Pros and Cons of Initial Coin Offering, Successful Initial Coin Offerings, Evolution of ICO, ICO Platforms.

UNIT-IV

Security in Blockchain: Introduction, Security Aspects in Bitcoin, Security and Privacy Challenges of Blockchain in General, Performance and Scalability, Identity Management and Authentication, Regulatory Compliance and Assurance, Safeguarding Blockchain Smart Contract (DApp), Security Aspects in Hyperledger Fabric.

Applications of Blockchain: Introduction, Blockchain in Banking and Finance, Blockchain in Education, Blockchain in Energy, Blockchain in Healthcare, Blockchain in Real-estate, Blockchain In Supply Chain, The Blockchain and IoT. Limitations and Challenges of Blockchain.

UNIT-V

Blockchain Case Studies: Case Study 1 – Retail, Case Study 2 – Banking and Financial Services, Case Study 3 – Healthcare, Case Study 4 – Energy and Utilities.
Blockchain Platform using Python: Introduction, Learn How to Use Python Online Editor, Basic Programming Using Python, Python Packages for Blockchain.
Blockchain platform using Hyperledger Fabric: Introduction, Components of Hyper ledger Fabric Network, Chain codes from Developer.ibm.com, Blockchain Application Using Fabric Java SDK.

Course Outcomes:

1. Understanding concepts behind crypto currency
2. Applications of smart contracts in decentralized application development
3. Understand frameworks related to public, private and hybrid blockchain
4. Create blockchain for different application case studies
5. Understand the security aspects in bitcoin using identity management and authentication.

TEXT BOOK:

1. "Blockchain Technology", Chandramouli Subramanian, Asha A. George, Abhilasj K A and Meena Karthikeyan, Universities Press.

REFERENCE BOOKS:

1. Michael Juntao Yuan, Building Blockchain Apps, Pearson, India.
2. Blockchain Blueprint for Economy, Melanie Swan, SPD O'reilly.
3. Blockchain for Business, Jai Singh Arun, Jerry Cuomo, Nitin Gaur, Pearson.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS64) SOFTWARE PROCESS & PROJECT MANAGEMENT
(Professional Elective – V)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

- To acquire knowledge on software process management.
- To acquire managerial skills for software project development.
- To understand software economics.

UNIT - I

Software Process Maturity

Software maturity Framework, Principles of Software Process Change, Software Process Assessment, The Initial Process, The Repeatable Process, The Defined Process, The Managed Process, The Optimizing Process, Process Reference Models Capability Maturity Model (CMM), CMMI, PCMM, PSP, TSP).

UNIT - II

Software Project Management Renaissance

Conventional Software Management, Evolution of Software Economics, Improving Software Economics, Life-Cycle Phases and Process artifacts

Engineering and Production stages, inception phase, elaboration phase, construction phase, transition phase, artifact sets, management artifacts, engineering artifacts and pragmatic artifacts, model-based software architectures.

UNIT - III

Workflows and Checkpoints of process

Software process workflows, Iteration workflows, Major milestones, minor milestones, periodic status assessments, Process Planning Work breakdown structures, Planning guidelines, cost and schedule estimating process, iteration planning process, Pragmatic planning.

UNIT - IV

Project Organizations

Line-of- business organizations, project organizations, evolution of organizations, process automation. Project Control and process instrumentation, The seven-core metrics, management indicators, quality indicators, life-cycle expectations, Pragmatic software metrics, metrics automation.

UNIT - V

CCPDS-R Case Study and Future Software Project Management Practices, Modern Project Profiles, Next-Generation software Economics, Modern Process Transitions.

Course Outcomes:

1. Understand the software process maturity, assessment, project plans and Quality Standards.
2. Examine the life cycle phases, artifacts, model-based software architecture.
3. Design workflows and checkpoints of a process.
4. Design and develop software project organizations and the seven-core metrics.
5. Identify the new project management process and practices.

TEXT BOOKS:

1. Managing the Software Process, Watts S. Humphrey, Pearson Education
2. Software Project Management, Walker Royce, Pearson Education

REFERENCE BOOKS:

1. An Introduction to the Team Software Process, Watts S. Humphrey, Pearson Education, 2000
2. Process Improvement essentials, James R. Persse, O'Reilly, 2006
3. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, TMH, 2006
4. Applied Software Project Management, Andrew Stellman & Jennifer Greene, O'Reilly, 2006.
5. Software Engineering Project Management, Richard H. Thayer & Edward Yourdon, 2nd edition, Wiley India, 2004.
6. Agile Project Management, Jim Highsmith, Pearson education, 2004.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS17) OPERATING SYSTEMS
(Open Elective –II)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Prerequisites:

1. A course on "Computer Programming and Data Structures".
2. A course on "Computer Organization and Architecture".

Course Objectives:

- Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection)
- Introduce the issues to be considered in the design and development of operating system
- Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

UNIT - I

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads

UNIT - II

CPU Scheduling - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

UNIT - III

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors **Interprocess Communication Mechanisms:** IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT - IV

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT - V

File System Interface and Operations -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

Course Outcomes:

1. Will be able to control access to a computer and the files that may be shared
2. Demonstrate the knowledge of the components of computers and their respective roles in computing.
3. Ability to recognize and resolve user problems with standard operating environments.
4. Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.
5. Ability to change access controls to protect files.

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
2. Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems- Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Crowley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education
5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS43) INTERNET OF THINGS
(Open Elective –II)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Pre-Requisites: Computer organization, Computer Networks

Course Objectives:

- To introduce the terminology, technology and its applications
- To introduce the concept of M2M (machine to machine) with necessary protocols
- To introduce the Python Scripting Language which is used in many IoT devices
- To introduce the Raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of web-based services on IoT devices

UNIT - I

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels and Deployment Templates

Domain Specific IoTs – Home automation, Environment, Agriculture, Health and Lifestyle

UNIT - II

IoT and M2M – M2M, Difference between IoT and M2M, SDN and NFV for IoT,

IoT System Management with NETCOZF, YANG- Need for IoT system Management, Simple Network management protocol, Network operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG

UNIT - III

IoT Systems – Logical design using Python-Introduction to Python – Python Data types & Data structures, Control flow, Functions, Modules, Packaging, File handling, Data/Time operations, Classes, Exception, Python packages of Interest for IoT

UNIT - IV

IoT Physical Devices and Endpoints - Raspberry Pi, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry PI with Python, Other IoT devices.

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Xively Cloud for IoT, Python

web application framework
–Django, Designing a RESTful web API

UNIT V

Case studies- Home Automation, Environment-weather monitoring-weather reporting- air pollution monitoring, Agriculture.

Course Outcomes:

1. Interpret the vision of IoT from a global context.
2. Perceive building blocks of Internet of Things and its characteristics.
3. Learn the basic concepts of Python. Implement python programming using Raspberry.
4. Perceive the application areas of IoT. Realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
5. Determine the Market perspective of IoT. Develop Python web applications and cloud servers for IoT.

TEXT BOOK:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547.

REFERENCE BOOK:

1. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS19) SOFTWARE ENGINEERING
(Open Elective –II)

B.Tech IV Year I Sem. **L T P C**
3 0 0 3

Course Objectives

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams.

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI)

Process models: The waterfall model, Spiral model and Agile methodology

UNIT - II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT - IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT - V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM

Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

Course Outcomes:

1. Define Software Engineering and list core principles of software engineering and understand various process models
2. Ability to translate end-user requirements into system and software requirements.
3. Implement to design UML and software architecture
4. Apply the testing strategies on different level of implementation.
5. Understand and able to compute quality measures and develop a software quality assurance plan for a software development.

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.

REFERENCE BOOKS:

1. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
2. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
4. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS65) CRYPTOGRAPHY AND NETWORK SECURITY LAB

B.Tech IV Year I Sem. **L T P C**
0 0 2 1

Course Objectives:

- Explain the objectives of information security
- Explain the importance and application of each of confidentiality, integrity, authentication and availability
- Understand various cryptographic algorithms.

List of Experiments:

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and display the result.
2. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms
   a. Ceaser cipher b. Substitution cipher c. Hill Cipher
4. Write a C/JAVA program to implement the DES algorithm logic.
5. Write a C/JAVA program to implement the Blowfish algorithm logic.
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Write the RC4 logic in Java Using Java cryptography; encrypt the text "Hello world" using Blowfish. Create your own key using Java key tool.
8. Write a Java program to implement the RSA algorithm.
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
11. Calculate the message digest of a text using the MD5 algorithm in JAVA

Course Outcomes:

1. Implement the cipher techniques.
2. Apply the mathematical foundation required for various cryptographic algorithms.
3. Develop the various security algorithms.
4. Use different open source tools for network security and analysis.

TEXT BOOKS:

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition
2. Cryptography and Network Security: Atul Kahate, McGraw Hill, 3rd Edition

REFERENCE BOOKS:

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition.
2. Cryptography and Network Security: Forouzan Mukhopadhyay, McGraw Hill, 3rd Edition
3. Information Security, Principles, and Practice: Mark Stamp, Wiley India.
4. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH
5. Introduction to Network Security: Neal Krawetz, CENGAGE Learning
6. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS66) COMPILER DESIGN LAB

B.Tech IV Year I Sem. **L T P C**
0 0 2 1

Prerequisites

1. A Course on "Object Oriented Programming through Java".

Co-requisites:

1. A course on "Web Technologies".

Course Objectives:

0. To understand the various phases in the design of a compiler.
1. To understand the design of top-down and bottom-up parsers.
2. To understand syntax directed translation schemes.
3. To introduce lex and yacc tools.

List of Experiments

1. Implementation of symbol table.
2. Develop a lexical analyzer to recognize a few patterns inc (ex. Identifiers, constants, comments, operators etc.)
3. Implementation of lexical analyzer using lex tool.
4. Generate yacc specification for a few syntactic categories.
   a) Program to recognize a valid arithmetic expression that uses operator +,-, * and /.
   b) Program to recognize a valid variable which starts with a letter followed by any number of letter or digits.
   c) Implementation of calculator using lex and yacc.
5. Convert the bnf rules into yacc form and write code to generate abstract syntax tree.
6. Implement type checking
7. Implement any one storage allocation strategies (heap, stack, static)
8. Write a lex program to count the number of words and number of lines in a given file or program.
9. Write a 'C' program to implement lexical analyzer using c program.
10. Write recursive descent parser for the grammar E->E+T E->T T->T*F T->F F->(E)/id.
11. Write recursive descent parser for the grammar
    S->(L)
    S->a L->L,S L->S
12. Write a C program to calculate first function for
    the grammar E->E+T E->T T->T*F T->F
    F->(E)/id
13. Write a YACC program to implement a top down parser for the given grammar.
14. Write a YACC program to evaluate algebraic expression.

Course Outcomes:

1. Apply client-server principles to develop scalable and enterprise web applications.
2. Design, develop, and implement a compiler for any language.
3. Use lex and yacc tools for developing a scanner and a parser.
4. Design and implement LL and LR parsers.

TEXT BOOK:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman.

REFERENCE BOOKS:

1. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Louden, Thomson.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS67)Project Stage - I

B.Tech IV Year I Sem.

L	**T**	**P**	**C**
0	0	6	3

Course Outcomes: On completion of the course, the students will be able to:

1. Identify the problem by applying acquired knowledge.
2. Analyze and categorize executable project modules.
3. Choose efficient tools for designing project modules.
4. Combine all the modules through effective team work after efficient testing.
5. Elaborate the completed task and compile the project report.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22MB09) ORGANIZATIONAL BEHAVIOR

B.Tech IV Year II Sem. **L T P C**

3 0 0 3

Course Objectives:

- This course demonstrates individual, group behavior aspects: The dynamics of organizational climate, structure and its impact on Organizations.

UNIT - I Organizational Behaviour

Definition, need and importance of organizational behaviour – Nature and scope – Frame work – Organizational behaviour models.

UNIT - II Individual Behaviour

Personality – types – Factors influencing personality – Theories – Learning – Types of learners – The learning process – Learning theories – Organizational behaviour modification, Misbehaviour – Types – Management Intervention. Emotions - Emotional Labour – Emotional Intelligence – Theories. Attitudes – Characteristics – Components – Formation – Measurement- Values. Perceptions – Importance – Factors influencing perception – Interpersonal perception- Impression Management. Motivation – importance – Types – Effects on work behavior.

UNIT - III Group Behaviour

Organization structure – Formation – Groups in organizations – Influence – Group dynamics – Emergence of informal leaders and working norms – Group decision making techniques – Team building - Interpersonal relations – Communication – Control.

UNIT - IV Leadership and Power

Meaning – Importance – Leadership styles – Theories of leadership – Leaders Vs Managers – Sources of power – Power centers – Power and Politics.

UNIT - V Dynamics of Organizational Behaviour

Organizational culture and climate – Factors affecting organizational climate – Importance. Job satisfaction – Determinants – Measurements – Influence on behavior. Organizational change – Importance – Stability Vs Change – Proactive Vs Reaction change – the change process – Resistance to change – Managing

change. Stress – Work Stressors – Prevention and Management of stress – Balancing work and Life. Organizational development – Characteristics – objectives –. Organizational effectiveness

Course Outcomes:

1. Explain need importance and structure of Organizational behavior.
2. Understands personality, perception and attitudes of Individuals in Organizations.
3. Determinates group behavioral aspects in organizations
4. Apply's various leadership theories and power politics.
5. Understands the dynamics of organizational climate.

TEXT BOOKS:

1. Stephen P. Robins, Organisational Behavior, PHI Learning / Pearson Education, 11th edition, 2008.
2. Fred Luthans, Organisational Behavior, McGraw Hill, 11th Edition, 2001.

REFERENCE BOOKS:

1. Schermerhorn, Hunt and Osborn, Organisational behavior, John Wiley, 9th Edition, 2008.
2. Udai Pareek, Understanding Organisational Behaviour, 2nd Edition, Oxford Higher Education, 2004.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS68) COMPUTATIONAL COMPLEXITY
(Professional Elective – VI)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Prerequisite: Design and Analysis of Algorithms.

Course Objectives:

- Introducing computational complexity-based algorithms and their implementations

UNIT – I

Introduction: Algorithms and complexity, Basic Complexity Classes-Deterministic time and the class P. Computational Tasks and models: Computational tasks – Search problems, Decision problems, Uniform models- Overview, General Principles, Concrete Model, Halting problem, restricted models.

UNIT – II

P vs. NP: Efficient Computation, The Search Version (Finding vs. Checking), The Decision Version (Proving Vs Verifying), Equivalence of the two formulations, Optimal Search Algorithms for NP Polynomial time reduction: The general notation of a Reduction, Reducing Optimization Problems to search problems, Self-Reducibility of search problems

UNIT – III

NP – Completeness: Definition, Cook's theorem, Existence of NP Complete Problems bounded halting and non-halting, Natural NP Complete Problems – The NP completeness of CSAT, The NP Completeness of SAT, Combinatorics and Graph Theory, additional properties of the standard reductions, Negative applications of NP Completeness, Positive applications of NP Completeness, NP Sets, Reflections on Complete problems, NP –complete optimization problems.

UNIT –IV

Diagonalization: Time Hierarchy theorem, Space Hierarchy theorem, Non-deterministic Time Hierarchy theorem, Ladner's theorem.

Space Complexity: Definition of space bounded computation, PSPACE completeness, NL Completeness, some space complexity classes– Savitch's theorem, Savitch's theorem, The essence of PSPACE

The polynomial time hierarchy and alternations: polynomial hierarchy, time versus alternations, properties of polynomial hierarchy, Complete problems in PH.

UNIT – V

Randomized computation: Probabilistic Turing machine, one sided and zero-sided error, Randomized reduction, Randomized space bounded computation.

Decision trees: Graphs and Decision Trees, Monotonic Graph properties, Topological criterion, Randomized decision trees.

Course Outcomes:

1. Understand the complexity of time and space for computational models.
2. Understand P and NP computation and Reduce optimization problems.
3. Able to learn and Implement NP completeness problems.
4. Understand hierarchical theorems and identifies membership and hardness of computational problems with respect to important complexity classes: P,NP,PSPACE,using their different characterizations.
5. Understand randomized computations and analyze the decision trees.

TEXT BOOKS:

1. The Basics of Computational Complexity, Oded Goldreich, Cambridge University Press
2. Computational Complexity: A Modern Approach, Sanjeev Arora and Boaz Barak, Princeton University

REFERENCE BOOKS:

1. Computational Complexity, by Christos Papadimitriou
2. Theory of Computational Complexity, Ding-Zhu Du, Ker-I Ko, WILEY

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS69) DISTRIBUTED SYSTEMS
(Professional Elective –VI)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Prerequisites:

1. A course on "Operating Systems".
2. A course on "Computer Organization & Architecture".

Course Objectives:

0. To provide an insight into Distributed systems.
1. To introduce concepts related to Peer to Peer Systems, Transactions and Concurrency control, Security and Distributed shared memory.

UNIT - I

Characterization of Distributed Systems: Examples of Distributed systems, Resource sharing and web, challenges

System models: Architectural and Fundamental models, Networking and Internetworking, Interprocess Communication

Distributed objects and Remote Invocation: Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

UNIT - II

Operating System Support- OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture.

Distributed File Systems-Introduction, File Service architecture.

UNIT - III

Peer to Peer Systems– Napster and its legacy, Peer to Peer middleware

Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging.

Coordination and Agreement- Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

UNIT - IV

Transactions and Concurrency Control- Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering.

Distributed Transactions-Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions

Distributed deadlocks: Transaction recovery.

UNIT - V

Replication: Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data.

Distributed shared memory: Design and Implementation issues, Consistency models.

Course Outcomes:

1. Understand of the principles and foundations on which the Internet and other distributed systems are based.
2. Apply different approaches for supporting distributed applications.
3. Analyze the role of middleware technologies in designing Distributed systems.
4. Understand Transactions and Concurrency control in distributed systems.
5. Analyze the sharing of data in distributed environment using various distributed algorithms.

TEXT BOOKS:

1. Distributed Systems Concepts and Design, G Coulouris, J Dollimore and T Kindberg, Fourth Edition, Pearson Education.
2. Distributed Systems, S. Ghosh, Chapman & Hall/CRC, Taylor & Francis Group, 2010.

REFERENCE BOOKS:

1. Distributed Systems – Principles and Paradigms, A.S. Tanenbaum and M.V. Steen, Pearson Education.
2. Distributed Computing, Principles, Algorithms and Systems, Ajay D. Kshemakalyani and Mukesh Singhal, Cambridge, rp 2010.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22AI20) DEEP LEARNING
(Professional Elective –VI)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Course Objectives:

- To understand deep Learning algorithms and their applications in real-world data

UNIT -I

Machine Learning Basics

Learning Algorithms, Capacity, Overfitting and Underfitting, Hyperparameters and Validation Sets, Estimators, Bias and Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning Algorithms, Unsupervised Learning Algorithms, Stochastic Gradient Descent, Building a Machine Learning Algorithm, Challenges Motivating Deep Learning

Deep Feedforward Networks Learning XOR, Gradient-Based Learning, Hidden Units, Architecture Design, Back-Propagation and Other Differentiation Algorithms

UNIT -II

Regularization for Deep Learning

Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under- Constrained Problems, Dataset Augmentation, Noise Robustness, Semi-Supervised Learning, Multi- Task Learning, Early Stopping, Parameter Tying and Parameter Sharing, Sparse Representations, Bagging and Other Ensemble Methods, Dropout, Adversarial Training, Tangent Distance, Tangent Prop, and Manifold Tangent Classifier, Optimization for Training Deep Models, Learning vs Pure Optimization, Challenges in Neural Network Optimization, Basic Algorithms, Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates

UNIT-III

Convolutional Networks

The Convolution Operation, Motivation, Pooling, Convolution and Pooling as an Infinitely Strong Prior, Variants of the Basic Convolution Function, Structured Outputs, Data Types, Efficient Convolution Algorithms, Random or Unsupervised Features

UNIT -IV

Recurrent and Recursive Nets

Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs,

Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Networks, Recursive Neural Networks, The Challenge of Long-Term Dependencies, Echo State Networks, Leaky Units and Other Strategies for Multiple Time Scales, The Long Short-Term Memory and Other Gated RNNs, Optimization for Long- Term Dependencies, Explicit Memory

UNIT -V

Practical Methodology: Performance Metrics, Default Baseline Models, Determining Whether to Gather More Data**,** Selecting Hyperparameters, Debugging Strategies, Example: Multi-Digit Number Recognition

Applications: Large-Scale Deep Learning, Computer Vision, Speech Recognition, Natural Language Processing, Other Applications.

Course Outcomes:

1. Understand machine learning basics and neural networks
2. Develop different parameters for Regularization for Deep Learning.
3. Apply CNN and RNN models for real-world data
4. Evaluate deep models
5. Develop deep models for real-world problems

TEXT BOOK:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.

REFERENCE BOOKS:

1. The Elements of Statistical Learning. Hastie, R. Tibshirani, and J. Friedman, Springer.
2. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press.
3. Bishop, C., M., Pattern Recognition and Machine Learning, Springer, 2006.
4. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
5. Golub, G., H., and Van Loan, C.,F., Matrix Computations, JHU Press, 2013.
6. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS70) HUMAN COMPUTER INTERACTION
(Professional Elective –VI)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Course Objectives:

- To gain an overview of Human-Computer Interaction (HCI)
- Understanding the alternatives to traditional "keyboard and mouse" computing.
- Getting familiarity with the vocabulary associated with sensory and cognitive systems
- Be able to apply models from cognitive psychology to predicting user performance
- Working in small groups on a product design with invaluable team-work experience.

UNIT - I

Introduction: Importance of user Interface – definition, importance of good design. Benefits of good design, A brief history of Screen design.

The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

UNIT - II

Design process – Human interaction with computers, importance of human characteristics, human consideration, Human interaction speeds, understanding business junctions.

Screen Designing: Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

UNIT- III

Windows – New and Navigation schemes selection of window, selection of devices based and screen- based controls. Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

UNIT- IV

HCI in the software process- The software life cycle, Usability engineering, Iterative design and prototyping, Design Focus: Prototyping in practice, Design rationale, Design rules, Principles to support usability Standards, Golden rules and heuristics, HCI patterns, Evaluation techniques, Goals of evaluation, Evaluation

through expert analysis, Evaluation through user participation, Choosing an evaluation method, Universal design, Universal design principles Multimodal interaction

UNIT- V

Cognitive models Goal and task hierarchies Design Focus: GOMS saves money, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures, Ubiquitous computing and augmented realities, Ubiquitous computing applications research, Design Focus: Ambient Wood – augmenting the physical, Virtual and augmented reality, Design Focus: Shared experience Design Focus: Applications of augmented reality Information and data visualization

Course Outcomes:

1. Understand the HCI and principles of GUI.
2. Design certain tools for blind or PH people.
3. Understand the Components of Human Computer Interaction.
4. Understand the importance of a design and evaluation methodology.
5. Understand the Hierarchical models to represent a user‘s task and goal.

TEXT BOOKS:

1. The essential guide to user interface design, Wilbert O Galitz, Wiley Dream Tech.
2. Human – Computer Interaction. Alan Dix, Janet Fincay, Gregory's, Abowd, Russell Bealg, Pearson Education.

REFERENCE BOOKS:

1. Designing the user interface. 3rd Edition Ben Shneidermann, Pearson Education Asia.
2. Interaction Design Prece, Rogers, Sharps. Wiley Dreamtech.
3. User Interface Design, Soren Lauesen, Pearson Education.
4. Human –Computer Interaction, D. R. Olsen, Cengage Learning.
5. Human –Computer Interaction, Smith - Atakan, Cengage Learning.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS71) CYBER FORENSICS
(Professional Elective –VI)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Prerequisites: Network Security.

Course Objectives:

- A brief explanation of the objective is to provide digital evidence which is obtained from digital media.
- In order to understand the objectives of computer forensics, first of all, people have to recognize the different roles computers play in a certain crime.
- According to a snippet from the United States Security Service, the computer functions in different kinds of crimes.

UNIT- I

Introduction of Cybercrime: Types, The Internet spawns crime, Worms versus viruses, Computers' roles in crimes, Introduction to digital forensics, Introduction to Incident - Incident Response Methodology – Steps - Activities in Initial Response, Phase after detection of an incident

UNIT- II

Initial Response and forensic duplication, Initial Response & Volatile Data Collection from Windows system -Initial Response & Volatile Data Collection from Unix system – Forensic Duplication: Forensic duplication: Forensic Duplicates as Admissible Evidence, Forensic Duplication Tool Requirements, Creating a Forensic. Duplicate/Qualified Forensic Duplicate of a Hard Drive

UNIT- III

Forensics analysis and validation: Determining what data to collect and analyze, validating forensic data, addressing data-hiding techniques, performing remote acquisitions

Network Forensics: Network forensics overview, performing live acquisitions, developing standard procedures for network forensics, using network tools, examining the honeynet project.

UNIT- IV

Current Forensic tools: evaluating computer forensic tool needs, computer forensics software tools, computer forensics hardware tools, validating and testing forensics

software E-Mail Investigations: Exploring the role of e-mail in investigation, exploring the roles of the client and server in e-mail, investigating e-mail crimes and violations, understanding e-mail servers, using specialized e-mail forensic tools.

Cell phone and mobile device forensics: Understanding mobile device forensics, understanding acquisition procedures for cell phones and mobile devices.

UNIT- V

Working with Windows and DOS Systems: understanding file systems, exploring Microsoft File Structures, Examining NTFS disks, Understanding whole disk encryption, windows registry, Microsoft startup tasks, MS-DOS startup tasks, virtual machines.

Course Outcomes:

1. Discuss forensics related to investigative process.
2. Explain the legal issues to prepare, perform forensic analysis based on the investigator's position.
3. Students will understand the usage of computers in forensic, and how to use various forensic tools for a wide variety of investigations.
4. Demonstrate the techniques, usage of forensics tools
5. It gives an opportunity to students to continue their zeal in research in computer forensics.

TEXT BOOKS:

1. Kevin Mandia, Chris Prosise, “Incident Response and computer forensics”, Tata McGraw Hill, 2006.
2. Computer Forensics, Computer Crime Investigation by John R. Vacca, Firewall Media, New Delhi.
3. Computer Forensics and Investigations by Nelson, Phillips Enfinger, Steuart, CENGAGE Learning

REFERENCE BOOKS:

1. Real Digital Forensics by Keith J. Jones, Richard Bejtiich, Curtis W. Rose, Addison- Wesley Pearson Education
2. Forensic Compiling, A Tractitioneris Guide by Tony Sammes and Brian Jenkinson, Springer International edition.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS72) ALGORITHMS DESIGN AND ANALYSIS
(Open Elective –III)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Prerequisites: Programming for problem solving and Data Structures

Course Objectives:
- Introduces the notations for analysis of the performance of algorithms.
- Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate;
- Describes how to evaluate and compare different algorithms using worst, average, and best case analysis.
- Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

UNIT - I

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

UNIT - II

Disjoint Sets: Disjoint set operations, union and find algorithms, Priority Queue-Heaps, Heapsort **Backtracking:** General method, applications, n-queen's problem, sum of subsets problem, graph Coloring, Hamiltonian cycles.

UNIT - III

Dynamic Programming: General method, applications- Optimal binary search tree, 0/1 knapsack problem, All pairs shortest path problem, Traveling sales person problem, Reliability design.

UNIT – IV

Greedy method: General method, applications- Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected components, Biconnected components.

UNIT – V

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem.

Course Outcomes:

1. Ability to analyze the performance of algorithms.Understand and apply divide and conquer algorithms.
2. Analyze the Disjoint set and Illustrate Backtracking algorithms for various applications.
3. Develop dynamic programming algorithms for various real-time applications
4. Understand and apply greedy methods, Implement non-linear data structure such as Trees, Graphs and its applications
5. Choose Branch and Bound, NP class problems and formulate solutions using standard approaches

TEXT BOOK:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharan, University Press.

REFERENCE BOOKS:

1. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education.
2. Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, PHI Pvt. Ltd./ Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R.Tamassia, John Wiley and sons.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS73) INTRODUCTION TO COMPUTER NETWORKS
(Open Elective – III)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Prerequisites
1. A course on "Programming for problem solving"
2. A course on "Data Structures"

Course Objectives
- Equip the students with the concepts and fundamentals of computer networks.
- Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

UNIT - I
Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.
Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless Transmission.
Data link layer: Design issues, framing, Error detection and correction.

UNIT - II
Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.
Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.
Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols, Data link layer switching.

UNIT - III
Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking

UNIT - IV
Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT - V
Application Layer: Domain name system, Electronic Mail; the World WEB, HTTP, Streaming audio and video.

Course Outcomes:

1. Gain the knowledge of the basics of computer networks and functions of each layer in the OSI and TCP/IP reference model.
2. Familiarity with the essential data link layer protocols, network design and implementation.
3. Design and evaluate subnetting, routing protocols to fulfill networking requirements.
4. Compare various congestion control algorithms and identify appropriate transport layer protocol for real time applications.
5. Analyze the features and operations of TCP/UDP, FTP, HTTP, SMTP,SNMP, etc.

TEXT BOOK:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 6th Edition. Pearson Education

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
2. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING

(B22CS74) INTRODUCTION TO CLOUD COMPUTING

(Open Elective – III)

B.Tech IV Year II Sem. **L T P C**
3 0 0 3

Prerequisites:

- A course on "Computer Networks"
- A course on "Operating Systems"
- A course on "Distributed Systems"

Course Objectives:

- This course provides an insight into cloud computing
- Topics covered include distributed system models, different cloud service models, service- oriented architectures, cloud programming and software environments, resource management.

UNIT-I

Computing Paradigms: High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing, Bio computing, Mobile Computing, Quantum Computing, Optical Computing, Nano computing.

UNIT-II

Cloud Computing Fundamentals: Motivation for Cloud Computing, The Need for Cloud Computing, Defining Cloud Computing, Definition of Cloud computing, Cloud Computing Isa Service, Cloud Computing Is a Platform, Principles of Cloud computing, Five Essential Characteristics, Four Cloud Deployment Models

UNIT-III

Cloud Computing Architecture and Management: Cloud architecture, Layer, Anatomy of the Cloud, Network Connectivity in Cloud Computing, Applications, on the Cloud, Managing the Cloud, Managing the Cloud Infrastructure Managing the Cloud application, Migrating Application to Cloud, Phases of Cloud Migration Approaches for Cloud Migration.

UNIT-IV

Cloud Service Models: Infrastructure as a Service, Characteristics of IaaS. Suitability of IaaS, Pros and Cons of IaaS, Summary of IaaS Providers, Platform as a Service, Characteristics of PaaS, Suitability of PaaS, Pros and Cons of PaaS, Summary of PaaS Providers, Software as a Service, Characteristics of SaaS, Suitability of SaaS, Pros and Cons of SaaS, Summary of SaaS Providers, Other Cloud Service Models.

UNIT-V

Cloud Service Providers: EMC, EMC IT, Captiva Cloud Toolkit, Google, Cloud Platform, Cloud Storage, Google Cloud Connect, Google Cloud Print, Google App Engine, Amazon Web Services, Amazon Elastic Compute Cloud, Amazon Simple Storage Service, Amazon Simple Queue ,service, Microsoft, Windows Azure, Microsoft Assessment and Planning Toolkit,

SharePoint, IBM, Cloud Models, IBM Smart Cloud, SAP Labs, SAPHANA Cloud Platform, Virtualization Services Provided by SAP, Sales force, Sales Cloud, Service Cloud: Knowledge as a Service, Rack space, VMware, Manjra soft, Aneka Platform

Course Outcomes:

1. Ability to understand various service delivery models of a cloud computing architecture.
2. Ability to understand the ways in which the cloud can be programmed and deployed.
3. Understand the Cloud Computing Architecture and Management.
4. Understand the cloud service model and its types.
5. Understand various cloud service providers.

TEXT BOOK:

1. Essentials of cloud Computing: K. Chandrasekhran, CRC press, 2014

REFERENCE BOOKS:

1. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Distributed and Cloud Computing, KaiHwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.
3. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O'Reilly, SPD, rp2011.

VAAGDEVI COLLEGE OF ENGINEERING
(Autonomous)
COMPUTER SCIENCE AND ENGINEERING
(B22CS75)Project Stage – II including Seminar

B.Tech IV Year II Sem.	**L**	**T**	**P**	**C**
Pre-requisites: -	0	0	22	11

Course Outcomes: At the end of this course, the students will be able to :

1. Identify the problem by applying acquired knowledge.
2. Analyze and categorize executable project modules.
3. Choose efficient tools for designing project modules.
4. Combine all the modules through effective team work after efficient testing.
5. Elaborate the completed task and compile the project report.