MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

B.Tech – Computer Science and Engineering - Artificial Intelligence & Machine Learning
COURSE STRUCTURE (R24)
Applicable From 2024-25 Admitted Batch
Structure Breakup

S.No	Category	Breakup of credits (Total 160 credits)
1	Humanities and social sciences including management courses (HSMC)	7
2	Basic Sciences Courses (BS)	20
3	Engineering sciences courses including workshop, drawing basics of electrical/mechanical/computer etc.(ES)	29
4	Professional core courses (PC)	58
5	Professional Electives (PE)	18
6	Open Electives (OE)	9
7	Project work, seminar and internship in industry or elsewhere (PS)	19
8	Mandatory Courses	-
	TOTAL	160

I YEAR I SEMESTER (I SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2410001	Matrices and Calculus	BS	3	1	0	4	40	60	100
2	2410008	Applied Physics	BS	3	0	0	3	40	60	100
3	2410501	Problem Solving using C and C++	ES	3	0	0	3	40	60	100
4	2410010	English for Skill Enhancement	HSMC	3	0	0	3	40	60	100
		Laboratory								
1	2410372	Engineering Workshop	ES	0	1	4	3	40	60	100
2	2410071	Applied Physics Laboratory	BS	0	0	2	1	40	60	100
3	2410571	Problem Solving using C and C++ Laboratory	ES	0	0	2	1	40	60	100
4	2410073	English Language and Communication Skills Laboratory	HSMC	0	0	2	1	40	60	100
		Skill Development Course								
1	2410596	Web Application Development	SDC	0	0	2	1	40	60	100
		Mandatory Course								
1		Foreign Language*	MC	0	0	0	0	-	-	-
		Induction Program								
Total Credits				**12**	**2**	**12**	**20**	**360**	**540**	**900**

- **Students can choose any one of the foreign language from the given list**
  - **i) 24X0FL1 French**
  - **ii) 24X0FL2 German**
  - **iii) 24X0FL3 Spanish**
  - **iv) 24X0FL4 Korean**

I YEAR II SEMESTER (II SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2420002	Differential Equations and Vector Calculus	BS	3	1	0	4	40	60	100
2	2420009	Engineering Chemistry	BS	3	0	0	3	40	60	100
3	2420201	Principles of Electrical and Electronics Engineering	ES	3	0	0	3	40	60	100
4	2420502	Essentials of Problem Solving using Python	ES	3	0	0	3	40	60	100
		Laboratory								
1	2420371	Computer Aided Engineering Graphics	ES	1	0	4	3	40	60	100
2	2420072	Engineering Chemistry Laboratory	BS	0	0	2	1	40	60	100
3	24X0271	Principles of Electrical and Electronics Engineering Laboratory	ES	0	0	2	1	40	60	100
4	2420572	Essentials of Problem Solving using Python Laboratory	ES	0	0	2	1	40	60	100
		Skill Development								
1	2420027	Public speaking skills	SDC	0	0	2	1	40	60	100
		Mandatory Course								
1	2420026	Yoga & Inner Engineering	MC	0	0	0	0	-	-	-
Total Credits				**13**	**1**	**12**	**20**	**360**	**540**	**900**

II YEAR I SEMESTER (III SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2436703	Discrete Mathematics	PC	2	0	0	2	40	60	100
2	2430510	Operating systems	PC	3	0	0	3	40	60	100
3	2430505	Software Engineering	PC	2	0	0	2	40	60	100
4	2430005	Mathematical and Statistical Foundations	BS	3	1	0	4	40	60	100
5	2430507	Data Structures	ES	3	0	0	3	40	60	100
	243ExL1	Design and Innovation	EL	0	0	2	1	40	60	100
		Laboratory								
1	2430574	Data visualization-Power BI	PC	0	0	2	1	40	60	100
2	2430577	Operating systems Lab	PC	0	0	2	1	40	60	100
3	2430575	Data Structures lab Using Python	ES	0	0	2	1	40	60	100
	2430588	Internship-1*	PS	0	0	2	1	100	-	100
		Skill Development								
1	2430598	Parallel Computation - RUST	SDC	0	0	2	1	40	60	100
		Mandatory Course								
1		Indian Knowledge System*	MC	0	0	0	0	-	-	-
Total Credits				**13**	**1**	**12**	**20**	**500**	**600**	**1100**

- **Students can choose any one of the following course**
  - i) 24XIKS1:Indian Science, Engineering and Technology
  - ii) 24XIKS2:Fundamentals and Applications of Vedic Mathematics
  - iii) 24XIKS3:Indian Health, Wellness and Psychology- including Ayurved
  - iv) 24XIKS4:Indian Town Planning and Architecture

II YEAR II SEMESTER(IV SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2440512	Artificial Intelligence	PC	3	0	0	3	40	60	100
2	2440503	Database Management Systems	PC	3	0	0	3	40	60	100
3	2446606	Computer System Architecture	PC	3	1	0	4	40	60	100
4	2440508	Design and Analysis of Algorithms	PC	2	0	0	2	40	60	100
5	2440511	OOPS through JAVA	ES	3	0	0	3	40	60	100
6	244EXL2	Prototype/model development and Entrepreneurship	EL	0	0	2	1	40	60	100
		Laboratory								
1	2440581	Artificial Intelligence Laboratory	PC	0	0	2	1	40	60	100
2	2440573	Database Management Systems Laboratory	PC	0	0	2	1	40	60	100
3	2440576	Design and Analysis of Algorithms through JAVA Laboratory	ES	0	0	2	1	40	60	100
		Skill Development								
1	2440597	NO SQL Data Bases (MONGO DB)	PS	0	0	2	1	40	60	100
		Mandatory Course								
1	2460021	Environmental Science	MC	0	0	0	0	-	-	-
Total Credits				14	1	10	20	400	600	1000

III YEAR I SEMESTER(V SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2450513	Computer Networks	PC	3	1	0	4	40	60	100
2	2450504	Formal Languages and Automata Theory	PC	3	0	0	3	40	60	100
3	2450515	Machine Learning	PC	3	0	0	3	40	60	100
4		Professional Elective-I	PC	3	0	0	3	40	60	100
5		Open Elective- I	OE	3	0	0	3	40	60	100
		Laboratory								
1	2450579	Computer Networks Laboratory	PC	0	0	2	1	40	60	100
2	2450585	Devops Laboratory	PC	0	0	2	1	40	60	100
3	2450583	Machine Learning Laboratory	PC	0	0	2	1	40	60	100
		Project								
1	2450587	Field Based Project *	PS	0	0	2	1	100	-	100
		Mandatory Course								
1	2450022	Gender Sensitization	MC	0	0	0	0	-	-	-
Total Credits				**15**	**1**	**8**	**20**	**420**	**480**	**900**

****students have to complete field based project in II year II semester summer break with minimum 2 weeks duration.**

III YEAR II SEMESTER (VI SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2460518	Data Mining	PC	3	0	0	3	40	60	100
2	2466601	Soft Computing	PC	3	1	0	4	40	60	100
3	2460517	Compiler Design	PC	3	1	0	4	40	60	100
4		Professional Elective-II	PE	3	0	0	3	40	60	100
5		Open Elective-II	OE	3	0	0	3	40	60	100
		Laboratory								
1	2466675	Augmented Reality and virtual Reality laboratory	PC	0	0	2	1	40	60	100
2	2466671	Soft Computing Laboratory	PC	0	0	2	1	40	60	100
3	2460586	Data Mining Laboratory	PC	0	0	2	1	40	60	100
		Mandatory Course								
1	2460023	Constitution of India	MC	0	0	0	0	-	-	-
Total Credits				**15**	**2**	**6**	**20**	**320**	**480**	**800**

IV YEAR I SEMESTER (VII SEMESTER)

S.No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
		Theory								
1	2470541	Deep Learning	PC	2	0	0	2	40	60	100
2	2470544	Full Stack Development	PC	3	0	0	3	40	60	100
3	2470010	Business Economics and financial Analysis	HSMC	3	0	0	3	40	60	100
4		Professional Elective-III	PE	3	0	0	3	40	60	100
5		Professional Elective-IV	PE	3	0	0	3	40	60	100
6		Open Elective-III	OE	3	0	0	3	40	60	100
		Laboratory								
1	2476672	Deep Learning Laboratory	PC	0	0	2	1	40	60	100
2	2470580	Full Stack Development Laboratory	PC	0	0	2	1	40	60	100
	2470588	**Internship-II****	PS	0	0	2	1	100	-	100
		Project								
1	2470590	Project Stage –I	PS	0	0	6	3	100	-	100
		Mandatory								
	2470025	Human Values and Professional Ethics	MC	0	0	0	0	-	-	-
Total Credits				**17**	**0**	**12**	**23**	**520**	**480**	**1000**

IV YEAR II SEMESTER (VIII SEMESTER)

S.No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
		Theory								
1		Professional Elective-V	PE	3	0	0	3	40	60	100
2		Professional Elective VI	PE	3	0	0	3	40	60	100
		Project								
1	2480589	Technical Seminar	PS	0	0	4	2	100	-	100
2	2480591	Project Stage–II	PS	0	0	18	9	40	60	100
Total Credits				**6**	**0**	**22**	**17**	**220**	**180**	**400**

PE I-Professional Elective I

S.No	Course Code	Course Title
1	24X6641	Ethical Hacking
2	24X0529	Data Science
3	24X6604	Artificial Neural Networks
4	24X0523	Internet of Things

PEII-Professional Elective II

S.No	Course Code	Course Title
1	24X0516	Cryptography Network security
2	24X6646	Information Retrieval Systems
3	24X6644	Digital Image Processing
4	24X0527	IoT Communication Protocols

PEIII– Professional Elective III

S.No	Course Code	Course Title
1	24X0528	Web Security
2	24X0533	Data Analytics
3	24X6603	Expert Systems
4	24X0531	IoT System Architectures

PEIV-Professional Elective IV

S.No	Course Code	Course Title
1	24X0540	Block chain Technology
2	24X0532	Cloud Computing
3	24X0538	Computer Vision & Robotics
4	24X6647	5G&IoTTechnologies

PEV-Professional Elective V

S.No	Course Code	Course Title
1	24X6645	Cyber Crime Investigation & Digital Forensics
2	24X0548	Social Network Analysis
3	24X0549	Quantum Computing
4	24X6652	Industrial IoT

PEVI-Professional Elective VI

S.No	Course Code	Course Title
1	24X0536	Cloud Security
2	24X0530	Natural Language Processing
3	24X0542	Cognitive Computing
4	24X6654	Generative AI

B.Tech – Computer Science and Engineering – (A I & M L)

1st Year (R24) Course Structure and Syllabus
Applicable From 2024-25 Admitted Batch

I YEAR I SEMESTER (I SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2410001	Matrices and Calculus	BS	3	1	0	4	40	60	100
2	2410008	Applied Physics	BS	3	0	0	3	40	60	100
3	2410501	Problem Solving using C and C++	ES	3	0	0	3	40	60	100
4	2410010	English for Skill Enhancement	HSMC	3	0	0	3	40	60	100
		Laboratory								
1	2410372	Engineering Workshop	ES	0	1	4	3	40	60	100
2	2410071	Applied Physics Laboratory	BS	0	0	2	1	40	60	100
3	2410571	Problem Solving using C and C++ Laboratory	ES	0	0	2	1	40	60	100
4	2410073	English Language and Communication Skills Laboratory	HSMC	0	0	2	1	40	60	100
		Skill Development Course								
1	2410596	Web Application Development	SDC	0	0	2	1	40	60	100
		Mandatory Course								
1		Foreign Language*	MC	0	0	0	0	-	-	-
		Induction Program								
Total Credits				**12**	**2**	**12**	**20**	**360**	**540**	**900**

- **Students can choose any one of the foreign language from the given list**
  - **i) 24X0FL1 French**
  - **ii) 24X0FL2German**
  - **iii) 24X0FL3Spanish**
  - **iv) 24X0FL4Korean**

I YEAR II SEMESTER (II SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2420002	Differential Equations and Vector Calculus	BS	3	1	0	4	40	60	100
2	2420009	Engineering Chemistry	BS	3	0	0	3	40	60	100
3	2420201	Principles of Electrical and Electronics Engineering	ES	3	0	0	3	40	60	100
4	2420502	Essentials of Problem Solving using Python	ES	3	0	0	3	40	60	100
		Laboratory								
1	2420371	Computer Aided Engineering Graphics	ES	1	0	4	3	40	60	100
2	2420072	Engineering Chemistry Laboratory	BS	0	0	2	1	40	60	100
3	2420271	Principles of Electrical and Electronics Engineering Laboratory	ES	0	0	2	1	40	60	100
4	2420572	Essentials of Problem Solving using Python Laboratory	ES	0	0	2	1	40	60	100
		Skill Development								
1	2420027	Public speaking skills	SDC	0	0	2	1	40	60	100
		Mandatory Course								
1	2420026	Yoga & Inner Engineering	MC	0	0	0	0	-	-	-
Total Credits				**13**	**1**	**12**	**20**	**360**	**540**	**900**

I-I

2410001: MATRICES AND CALCULUS

(CSE, CSD, CSM, ECE, EEE, MECH, CIVIL)

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

3 1 0 4

Course Overview:

Matrix algebra and calculus are essential for understanding and solving complex problems in many scientific and engineering fields. This course provides the mathematical foundation for advanced topics and applications. This course covers matrix theory, linear algebra and calculus. Linear algebra is a branch of mathematics that studies system of linear equations and the properties of matrices. The calculus part of the course typically covers differential and its applications, and integration techniques. Matrix algebra allows us to think of a matrix holistically, generalize and compute derivatives important matrix factorizations, understand how differentiation formulas must be reimagined in large scale computing. Calculus is used to model and solve real- world problems.

Pre-requisites: Mathematics courses of 10+2 year of study.

Course Objectives: The student will try 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, eigenvectors and reduction of quadratic form to canonical form by orthogonal transformation.
- Geometrical approach to the mean value theorems and their application to the mathematical problems. Evaluation of improper integrals using Beta and Gamma functions.
- Partial differentiation, concept of total derivative and finding maxima and minima of function of two and three variables.
- Evaluation of multiple integrals and their applications.

Course outcomes: After successful completion of the course, students should be able to

CO1: Write the matrix representation of a set of linear equations and to analyses the solution of the system of equations.

CO2: Find the Eigen values and Eigenvectors and reduce the quadratic form to canonical form using orthogonal transformations.

CO3: Solve the applications on mean value theorems and evaluate improper integrals using Beta and Gamma functions.

CO4: Find the extreme values of functions of two variables with/ withoutconstraints.

CO5: Evaluate the multiple integrals and apply the concept to find areas, volumes.

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, L-Udecomposition method.

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

Eigen values, Eigen vectors and their properties (without proof), Diagonalization of a matrix, Cayley-Hamilton Theorem (without proof), finding inverse and power of a matrixbyCayley-HamiltonTheorem,QuadraticformsandNatureoftheQuadraticForms,Reductionof Quadratic form to canonical forms by Orthogonal Transformation.

UNIT-III:Calculus **8 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 (without proofs).

Beta and Gamma functions and their applications(properties without proof).

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

Partial Differentiation: Euler's Theorem, Total derivative, Jacobian, Functional dependence-independence.Applications:Maximaandminimaoffunctionsoftwovariablesandthreevariables using method of Lagrangemultipliers.

UNIT-V: Multi variable Calculus (Integration) **10 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 triple integrals (Cartesian to Spherical and Cylindrical polar coordinates).

Applications: Areas (by double integrals) and volumes (by triple integral).

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, 5thEditon,2016.

REFERENCE BOOKS:

1. Erwin kreyszig, Advanced Engineering Mathematics, 9thEdition, John Wiley & Sons,2006.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9thEdition,Pearson, Reprint, 2002.
3. H. K. Dassand Er. RajnishVerma, Higher Engineering Mathematics, S Chand and Company Limited, NewDelhi.

2420008: APPLIED PHYSICS

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

3 1 0 4

Course Overview

Applied Physics is the application of the Physics to solve Scientific or Engineering Problems. It is considered as bridge between Physics and Engineering.

Applied Physics under graduate program stress the basic Physics that underlies most developments in engineering and mathematical tools that are important to engineers.

Prerequisites: 10 + 2 Physics

Course Objectives: The student will try to learn :

1. Understand the basic principles of quantum physics and band theory of solids.
2. Understand the underlying mechanism involved in construction and working principles of various semiconductor devices.
2. Study the fundamental concepts related to the dielectric, magnetic materials.
3. Identify the importance of nanoscale, quantum confinement and various fabrications techniques.
5. Study the characteristics of lasers and optical fibers.

Course Outcomes: After successful completion of the Course the students should be able to

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.
2. Identify the role of semiconductor devices in science and engineering Applications.
3. Explore the fundamental properties of dielectric, magnetic materials and their applications.
4. Appreciate the features and applications of Nano materials.
5. Understand various aspects of Lasers and Optical fiber and their applications in diverse fields.

UNIT - I: QUANTUM PHYSICS AND SOLIDS

Quantum Mechanics: Introduction to quantum physics, Blackbody radiation, Photoelectric effect, de-Broglie Hypothesis and 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), Bloch's theorem -Kronig-Penney model, 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 of operation and characteristics of P-N Junction diode, Zener diode and Bipolar junction transistor (BJT)

Opto-devices- Light emitting diode (LED), PIN diode, and Solar cell, their structure, materials, working principle and characteristics, Solar cell application- Space craft.

UNIT - III: DIELECTRIC AND MAGNETIC MATERIALS

Dielectric Materials: Introduction to dielectrics, Polarization, Permittivity, Dielectric constant, Types of polarizations (Qualitative), Internal field in Solids, Clausius-Mossotti equation, Ferroelectric, Piezoelectric and Pyroelectric materials, Applications.

Magnetic Materials: Introduction to Magnetism, Magnetization, Permeability, Susceptibility, Classification of Magnetic Materials, Hysteresis curve, Soft and Hard magnetic materials, Magnetostriction, Magneto resistance, Magnetic field sensors and bubble memory devices.

UNIT - IV: NANOTECHNOLOGY

Nanoscale, Quantum confinement, Surface to volume ratio, Bottom-up fabrication: Sol-gel, Precipitation methods, Top-down fabrication: Ball milling, Physical Vapor Deposition (PVD), Characterization techniques - XRD, SEM and TEM, Applications of nanomaterials.

UNIT - V: LASER AND FIBER OPTICS

Lasers: Laser beam characteristics-three quantum processes-Einstein coefficients and their relations, Lasing action, Population inversion, Pumping methods, Ruby laser, He-Ne laser, CO_2 laser, Applications of laser- Medical and Military.

Fiber Optics: Introduction to optical fiber, 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 of optical fiber-Endoscopy.

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, McGraw 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, 1stEdition, 2021.

REFERENCE BOOKS:

1. Quantum Physics, H.C. Verma, TBS Publication, 2ndEdition 2012.
2. Fundamentals of Physics – Halliday, Resnick and Walker, John Wiley &Sons,11thEdition, 2018.
3. Introduction to Solid State Physics, Charles Kittel, Wiley Eastern, 2019.
4. Elementary Solid State Physics, S.L. Gupta and V. Kumar, Pragathi Prakashan, 2019.
5. A.K. Bhandhopadhya - Nano Materials, New Age International, 1stEdition, 2007.

24X0501:Problem Solving Using C and C++

B.Tech. I Year. I Sem.

L	T	P	C
3	0	0	3

Course Overview:

The Course Provides good foundation in procedural oriented and object-oriented programming concepts. It provides overview on basic building blocks of procedural oriented concepts like arrays, pointers, structures, strings**.** It comprises object-oriented concepts such as information hiding, encapsulation, inheritance and polymorphism. C programming is used in operating systems, embedded devices, OS kernels, drivers, IoT applications. C++ is widely used for creating graphics-heavy software, game engines, VR applications, and web browsers.

Prerequisites: Nil

Course Objectives: The students will try to learn

- Using of structured programming approach in solving problems
- How to use arrays ,pointers, strings and structures in solving problems
- Defining of structures in C and classes in C++
- Importance of inheritance in object-oriented programming
- Handling of exceptions in programs

Course Outcomes: After successful completion of the course, students should be able to

- Develop programs using Control statements and Repetitive statements
- Modularize the code with functions so that they can be reused
- Learn about Object oriented concepts
- Design programs by using Inheritance concepts
- Implement polymorphism and Exception Handling

Module-I:Introductiontoprogramming [10]

Introduction Procedure Oriented and Object-Oriented Programming. Algorithm, Flowchart, Pseudo code. Creating and Running of C Program. Structure of C program – C character set, C Tokens: Constants, Variables, Keywords, Identifiers, C data types, C operators. Standard I/O in C (scanf, printf), Conditional Control statements (if and Switch) Statements. Repetitive statements: While, Do While and For Loops - Use of Break and Continue Statements.

Module-II: Functions, Arrays, Strings and Pointers [12]

Arrays:Introduction,Declaration,CreatingandAccessingofOne-Dimensional Arrays, Two- Dimensional Arrays.
StringsandPointers:Introductiontostrings,stringhandlingfunctions,Arraysof strings, Introduction to pointers, Dynamic Memory allocation.
Functions: Defining Functions – User Defined Functions, Storage Classes, passing parameters:CallByValue,CallByReference,Recursion,Command-lineArguments.

Module-III: Structures and Classes **[8]**

Structures: Defining structures, initializing structures, unions, Array of structures
OOPS Concepts: Class, Object, Abstraction, Encapsulation, Inheritance and Polymorphism.
C++ Classes and Data Abstraction: Class definition, Class structure, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Dynamic creation and destruction of objects, Data abstraction.

Module–IV: Inheritance **[7]**

Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class.

Module–V: Polymorphism and ExceptionHandling **[8]**

Virtual Functions and Polymorphism: Static and Dynamic binding, virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Implications of polymorphic use of classes, Virtual destructors.
Exception handling: Try, throw and catch.

TEXTBOOKS:

1. Forouzan B.A&RichardF.Gilberg,AStructuredProgrammingApproach using C,3rd Edition(2013), Cengage Learning.
2. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Designin C 7th Edition,Pearson
3. ANSIandTurboC++byAshokeN.Kamthane,PearsonEducation
4. RobertLaforeC++

REFERENCES:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. E.Balagurusamy,ComputerfundamentalsandC,2ndEdition,McGraw-Hill
3. YashavantKanetkar,LetUsC,18thEdition,BPB
4. E.Balagurusamy, Object Oriented Programming using C++,2ndEdition, McGraw-Hill

24X0010: ENGLISH FOR SKILL ENHANCEMENT

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

3 0 0 3

Course Overview:

1. The English language plays a vital role in engineering education. Acquiring LSRW skills has become a prerequisite to learning about different technologies and their intricacies.
2. All these extracts are fascinating, thought-provoking, and contextual to engineering students. The authors have sincerely tried connecting every lesson with the modules of vocabulary, grammar, reading comprehension and writing tasks stipulated under each module. The textbook includes several exercises and activities involving the student's language skills practice. They are extremely encouraging and motivational and cater to a group of students with mixed abilities.
3. Each module starts with the preparatory task which can stimulate an interesting discussion among the students in the classroom. Adequate explanations and more examples are provided in vocabulary and grammar sections to enable students to work independently in and outside the classroom. The reading part suggests improving students' reading skills and provides reading comprehension exercises. The writing module aims at developing the learner's writing skills by providing conceptual discussions and exercises in different forms of written communication such as formal letters, CV/ résumé and job application letters, e-mails, reports, etc.

Prerequisites: Language Comprehension

Course Objectives: The students will try to learn:

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

Course Outcomes: After successful completion of the course, students should be able to:

1. Understand the importance of vocabulary and sentence structures.
2. Choose appropriate vocabulary and sentence structures for their oral andwritten communication.
3. Demonstrate their understanding of the rules of functional grammar.
4. Develop comprehension skills from the known and unknown passages.
5. Take an active part in drafting paragraphs, letters, essays, abstracts, précis andreports in various contexts.

MODULE – I (No of Hours = 7)

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

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 concerning Articles and Prepositions.

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

Writing: Sentence Structures -Use of Phrases and Clauses in 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.

MODULE – II (No of Hours = 6)

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

Vocabulary: Words Often Misspelt - Homophones, Homonyms and Homographs

Grammar: Identifying Common Errors in Writing concerning Noun-pronounAgreement and Subject-verb Agreement.

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.

MODULE – III (No of Hours = 7)

The chapter entitled **'Lessons from Online Learning' by F.Haider Alvi, Deborah Hurst et al** from

"English: Language, Context and Culture" published by Orient BlackSwan, Hyderabad.**Vocabulary**: Words Often Confused - Words from Foreign Languages and their Use in English. **Grammar:** Identifying Common Errors in Writing Concerning 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.

MODULE – IV (No of Hours = 6)

Chapter entitled **'Art and Literature' by Abdul Kalam** from ***"English: Language, Contextand Culture"*** published by Orient BlackSwan, Hyderabad.

Vocabulary: Standard Abbreviations in English

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.

MODULE – V (No of Hours = 6)

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

Vocabulary: Technical Vocabulary and their Usage

Grammar: Common Errors in English (*Covering all the other aspects of grammar whichwere not covered in the previous modules*)

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 Module 6 in the AICTE ModelCurriculum are covered in the syllabus of the 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 **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 in adopting Blended Learning in dealing with the course contents. They are advised to teach 40 per cent of each topic from the syllabus in blended mode.

TEXTBOOK:

1. "English: Language, Context and Culture" by Orient BlackSwan Pvt. Ltd, Hyderabad. 2022. Print.

REFERENCE BOOKS:

1. Effective Academic Writing by Liss and Davis (OUP)
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 FunctionalLanguage, Grammar, and Vocabulary. (2nd ed.,). Sage Publications India Pvt. Ltd.
5. (2019). Technical Communication. Wiley India Pvt. Ltd.
6. Vishwamohan, Aysha. (2013). English for Technical Communication forEngineering Students. Mc Graw-Hill Education India Pvt. Ltd.
7. Swan, Michael. (2016). Practical English Usage. Oxford University Press. Fourth Edition.

24X0372: ENGINEERING WORK SHOP
(Common to all branches)

B.Tech.I Year.I Semester

L	T	P	C
0	1	4	3

Course Overview: After successful completion of the course, students should able to learn.

Engineering Workshops: A foundational course aimed at introducing first-year students to a variety of tools, equipment, and techniques essential for creating physical objects and mechanisms using different materials. This course provides an opportunity for students to build confidence and gain practical experience in carpentry, fitting, house wiring, tin-smithy, black smithy, welding, and principles of modern manufacturing processes.

Prerequisite: NIL

Course Objective: The student will able to

1. To gain good basic working knowledge required for the production of various engineering products.
2. To study different tools uses and their demonstration.
3. To provide hands on experience about use of different engineering materials, tools, equipments and processes those are common in engineering field.
4. To develop a right attitude, team working, precision and safety at work place.
5. It explains the construction, function use and application different working tools and equipments.

Course outcomes: At the end of the course students should able to

1. Explain the design and model different prototype in the trade of carpentry such as Cross lap joint, Dove tail joint.
2. Demonstrate the design and model various basic prototypes in the trade of fitting such as straight fit and V-fit.
3. Understand to make various basic prototypes in the trade of tin smithy such as rectangular tray and open cylinder.
4. Demonstrate the design and model various basic prototype in welding and black smithy.
5. Understand to perform various basic house wiring techniques such as connecting one lamp with one switch, connecting two lamps with one switch.

Course content:

Module -I: CARPENTRY AND FITTING

- Carpentry – Introduction, Carpentry tools, sequence of operations and applications (T-Lap Joint, Dovetail Joint, Mortise & Tenon Joint)
- Fitting – Introduction, fitting tools, sequence of operations and applications (V-Fit, Dovetail Fit & Semi-circular fit)

Module -II: TIN SMITHY AND BLACKSMITHY

- Tin-Smithy – Introduction, Tin smithy tools, sequence of operations and applications (Square Tin, Rectangular Tray & Conical Funnel).
- Blacksmithy- Introduction, Blacksmithy tools, sequence of operations and applications (Round to Square, Fan Hook and S-Hook)

Module -III: HOUSE WIRING AND WELDING

- House-wiring – Introduction, Electrical wiring tools, sequence of Operations and applications (Parallel & Series, Two-way Switch and Tube Light)
- Welding Practice – Introduction, electrode, welding tools, and sequence of Operations. Advantages and applications (Arc Welding).

Text Book:

1. Workshop practice/B.L. Juneja/ Cengage
2. Workshop manual/K.Venugopal/ Anuradha

Reference:

1. Work shop manual /P.Kannaiah/K.L. Narayana
2. Work shop Manual /Venkat Reddy /BSP

24X00071: APPLIED PHYSICS LAB

B.Tech. I Year I Sem.

L	T	P	C
0	0	3	1.5

Course Objectives: The objectives of this course for the student to

1. Capable of handling instruments related to the Hall effect and photoelectric effect Experiment understands their measurements.
2. Understand the characteristics of various devices such as PN junction diode, Zener diode, BJT, LED, solar cell, lasers and optical fiber and measurement of energy gap.
3. Apply the analytical techniques & graphical analysis for Stewart Gees , LCR & RC .
4. Understanding the method of least squares fitting.
5. To develop intellectual communication skills through discussion on basic principles of scientific concepts in a group.

Course Outcomes: The students will be able to:

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.
2. Appreciate quantum physics in semiconductor devices and optoelectronics.
3. Gain the knowledge in calculating the quality factor and time constant of LCR and RC circuits.
4. Understand the variation of magnetic field at various points.
5. Carried out data analysis.

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. V-I and L-I characteristics of light emitting diode (LED) and LASER.
7. V-I Characteristics of solar cell.
8. Determination of Energy gap of a semiconductor.
9. To determine the time constant of R-C circuit.
10. Determination of Acceptance Angle and Numerical Aperture of an optical fiber.
11. Understanding the method of least squares – Torsional pendulum as an example.
12. Determination of magnetic field induction along the axis of a current carrying coil.

REFERENCE BOOK:

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

24X0571:Problem Solving Using C and C++ Laboratory

B.Tech. I Year. I Sem

L	T	P	C
0	0	2	1

Course Overview:

The Course Provides good foundation in procedural oriented and object-oriented programming concepts. It provides overview on basic building blocks of procedural oriented concepts like arrays, pointers, structures, strings**.** It comprises object-oriented concepts such as information hiding, encapsulation, inheritance and polymorphism. C programming is used in operating systems, embedded devices, OS kernels, drivers, IoT applications. C++ is widely used for creating graphics-heavy software, game engines, VR applications, and web browsers.

Prerequisites: Nil

Course Objectives: The students will try to learn

- Using of structured programming approach in solving problems
- How to use arrays, pointers, strings and structures in solving problems
- Defining of structures in C and classes in C++
- Importance of inheritance in object-oriented programming
- Handling of exceptions in programs

Course Outcomes: After successful completion of the course, students should be able to

- Develop programs using Control statements and Repetitive statements
- Modularize the code with functions so that they can be reused
- Learn about Object oriented concepts
- Design programs by using Inheritance concepts
- Implement polymorphism and Exception Handling

Week1: Algorithm and Flowchart

1. You are designing a flowchart and algorithm for a distance and speed calculator. The flowchart should prompt the user to input the distance traveled and the time taken and calculate the speed using the formula: **Speed = Distance / Time**.

2. You are developing an algorithm and flow chart for a circle area calculator. The flowchart should prompt the user to input the radius of a circle and calculate the area using the formula: **Area = πr^2**

3. You are designing an algorithm and flowchart for a fuel efficiency calculator in a car rental app. The flowchart should prompt the user to input the distance traveled and the amount of fuel consumed and calculate the fuel efficiency in miles per gallon (MPG) using the formula MPG = Distance / Fuel Consumption.

4. You are developing an algorithm and flowchart for a discount calculator in an online shopping app. The flowchart should prompt the user to input the original price and the discount percentage and calculate the discounted price using the formula: **Discounted Price = Original Price - (Original Price * Discount Percentage /100).**

5. You are developing a flowchart and algorithm for a monthly budget tracker. The flowchart should prompt the user to input their income and expenses and calculate the total savings using the formula **Total Savings = Income - Total Expenses**.

6. You are designing a flowchart for a loan amortization calculator. The flowchart should prompt the user to input the loan amount, interest rate, and duration, and calculate the monthly payment using the formula Monthly Payment = (Loan Amount * Interest Rate * $(1+ \text{Interest Rate})^{\text{Duration}}$) / ($(1 + \text{Interest Rate})^{\text{Duration-1}}$).

7. You are developing a flowchart and algorithm for a construction materials calculator. The Flow chart should guide the user through the process of inputting the dimensions ofa room, including the length, width, and height. The goal is to calculate the surface area of the room using the provided formula:
   Surface Area=2*(Length*Width+Length *Height+Width* Height).

8. Developaflowchartandanalgorithmtoconvertagiventimeinhoursandminutesto minutes only. Prompt the user to input the time in hours and minutes and display the converted time in minutes.

Skill Oriented Exercise

9. Farmer Thimmayya bought some mules at Rs. 50 each, sheep at Rs. 40 each, goats at Rs. 25each,andpigsatRs.10each.Theaveragepriceoftheanimalsperheadworkedto Rs. 30.What is the minimum number of animals of each kind did he buy?

10. **AMatterofRupeesandPaisa:** Ihavemoneypouchcontaining Rs.700.Thereareequal numberof25 paise coins, 50 paise coins and 1 rupee coins. How manyof each are there.

11. Developanalgorithmandflowchartthatpromptstheusertoinputtheinitialvelocity, acceleration, and time. Calculate and display the final velocity using the formula Final Velocity = Initial Velocity + (Acceleration * Time).

12. Develop an algorithm and flowchart that prompts the user to input the lengths of the three sides of a triangle. Calculate and display the area of the triangle using Heron's formula: Area=$\sqrt{}$(s*(s -Side1)*(s-Side2)*(s -Side3)),where s=(Side1+Side2+ Side3) / 2.

Week2: Algorithm and Flowchart

1. ABC Company wants to calculate the monthly salary for its employees based on various components such as basic pay, DA, HRA, and deductions for taxes and provident fund. The company follows the following rules for salary calculation:
   - The basic pay is a fixed amount each employee receives.
   - DAiscalculatedas20 percentage of the basic pay.
   - HRA is calculated as 10percentageofthebasicpay.
   - The gross salary is the sum of the basic pay, DA, and HRA.
   - The net salary is the gross salary minus deductions for taxes and provident fund.

   Develop an algorithm and flow chart to calculate the gross and net salary of the employee. Include the necessary steps to calculate the net salary.

2. Sarah, a dedicated student, wants to calculate her average grade for a semester. She has received marks in four different subjects and seeks assistance in creating an algorithm to determine her average grade based on these marks. Develop an algorithm and flowchart to help Sarah calculate her average grade for the semester using the marks obtained in these four subjects. Ensure that the algorithm includes the necessary steps to compute the average grade accurately.

3. John is a programmer who wants to convert a given temperature in Celsius to Fahrenheit. Help John in developing an algorithm and flowchart to perform this conversion. Design an algorithm for John to convert a given temperature in Celsius to Fahrenheit. Provide step-by-step instructions for performing the conversion.

4. **The Tall Men Next Door:** Next door to me live four brothers of different heights. Their average height is 74 inches, and the difference in height among the first three men is two inches. The difference between the third and the fourth man is six inches. Can you tell how tall each brother is?

5. **Driving Through the Country:** I decided to travel through the countryleisurely and on the first day I did only 7 miles. On the last day I did 51 miles, increasing my journey by 4 miles per day. How many days did I travel and how far?

6. There is a beautiful pond in a park, filled with clear water. The park management wants to monitor the water level in the pond regularly to ensure it remains at an optimal level. They have asked you to create an algorithm to help them with this task. Develop an algorithm and flow chart to monitor the water level in the pond and notify the park management if it falls below a certain threshold. (Note: You can assume that the input for the current water level and threshold level is obtained from a monitoring device or sensor).

7. If a participant can make one submission every 45 seconds, and a contest lasts for Y minutes, create an algorithm and flowchart to find the maximum number of submissions that the participant can make during the contest? Assume the participant is allowed to make submissions until the last second of the contest.

Skill Oriented Exercise

8. Michael wants to find the largest number among a set of given numbers. Help Michael write an algorithm to determine the largest number from the given inputs. Create an algorithm and flowchart for Michael to find the largest number among a set of given inputs. Provide step-by-step instructions to identify the largest number.

9. A construction worker needs to paint the exterior walls of a rectangular building. The dimensions of the walls are L meters in length, H meters in height, and W meters in width. If the cost of painting is Rs. 20 per square meter, what will be the total cost of painting the walls? Prepare an algorithm and flowchart to calculate the total cost of painting.

10. An ice cream vendor brings 'i' litre of ice cream to a fair. Each cone requires0.25liters of ice cream. If the vendor sells 80 cones, Develop an algorithm and flowchart to find the number of liters of ice cream left with the vendor.

11. Amanda is planning a party and wants to determine the total number of guests attending. Assist Amanda by writing an algorithm to calculate the total number of guests based on the number of adults and children invited. Create an algorithm and flowchart for Amanda to calculate the total number of guests attending her party, considering the number of adults and children invited. Include the necessary steps to calculate the total number of guests.

Week: 3 Data Types, Console I/O, Operators Storage Classes

1. You are working as a financial analyst at a bank. As part of your job, you need to calculate the maturity amount for fixed deposits (FD) based on the principal amount, interest rate, and duration. For that help me to write a C program that takes the principal amount, interest rate, and duration(in years)as input from the user. Calculate and display the maturity amount using the simple interest formula: Maturity Amount = Principal + (Principal * Interest Rate * Duration).

2. Jenny, a budding mathematician, was studying the concept of area and perimeter. She was given a rectangular garden with a known length and width. Jenny wondered if she could find the area without knowing the width. Can you help Jenny derive a formula to calculate the area of a rectangle using only the length and perimeter?

3. Develop a C program that computes the hypotenuse of a right-angled triangle given the lengths of its two perpendicular sides. Prompt the user to enter the lengths and display the result. (Pythagorean theorem:$Hypotenuse^2 = Side1^2 + Side2^2$)

4. Once upon a time, there was a mathematician named Alex. Alex loved solving mathematical problems and puzzles. One day, Alex came across an ancient scroll that contained a secret formula to calculate the sum of the first n natural numbers. The scroll mentioned that by using the formula, one could find the sum of any given number of natural numbers without having to manually add them up. Alex was intrigued and decided to test the formula. Can you help Alex implement a C program that uses the formula to calculate the sum of the first n natural numbers?

Instructions:

Write a C program that takes an input integer n from the user and calculates the sum of the first n natural numbers using the formula: **sum=(n*(n+1))/2**

5. Once upon a time in a small town, there were two friends, Kavi and Jei, who were fascinated by the concept of slopes in mathematics. They loved exploring the hills and valleys around their town and wondered how they could calculate the slope of any given landscape. One sunny day, while hiking up a hill, Kavi and Jei discovered an ancient map that had the secret to finding the slope of a line between two points. The map indicated that by using the coordinates of two points, they could determine the slope of the line connecting them. Excited about their discovery, Kavi and Jei decided to create a C program that could calculate the slope for any two points. They wanted to share their program with others so that everyone could explore the slopes of various land scapes. Can you help Kavi and Jei bring their idea to life by implementing a C program that calculates the slope of a line?

Instructions:

Write a C program that prompts the user to enter the coordinates of two points:(x1,y1)and (x2, y2). The program should then

Calculate ,the slope of the line connecting these points using the formula: slope=(y2-y1) /(x2- x1)

Finally, the program should display the calculated slope to the user.

6. Ravi and Kavi are on an exciting treasure hunt adventure, following a map with hidden treasures located at different coordinates. They want to calculate the distance between two treasures to determine how far apart they are. Can you help them by writing a C program that performs this calculation?

Instructions:

Write a C program that prompts the user to enter the coordinates of two treasures: Treasure A and Treasure B. The coordinates should be in the form (x, y). Calculate the distance between the treasures using the distance formula:

distance = sqrt ((x2 - x1)^2 + (y2 - y1)^2)

Finally ,output a message indicating how far a part the treasures are.

7. Imagine a scenario where there is a coconut tree with multiple coconuts hanging from it. There is a person standing at a distance of "D" meters away from the tree. The coconuts are positioned at a height of "H" meters from the ground. Could you please help write a C program that calculates the angle at which the person should aim in order to hit the coconuts? (Hint: you can use the inverse tangent function (atan() in C) to determine the angle based on the ratio of the height of the coconuts to the distance from the tree.)

Skill Oriented Exercise

8. In a faraway kingdom, two treasure hunters named Alex and Bella embarked on a daring quest to find valuable treasures. While exploring a mysterious cave, they stumbled upon two treasure chests, each containing a unique gemstone. Curiosity took over, and they decided to swap the gemstones inside the chests. However, a magical enchantment prevented them from directly swapping the gemstones. To fulfill their quest and restore the treasures to their rightful chests, Alex and Bella realized they could use a third variable and arithmetic operations. Can you help them by writing a C program that takes the values of the gemstones as input, and swap their values.

9. Emily, a young architect, was working on designing a cylindrical water tank for a new building. As she was finalizing the plans, she needed to calculate the surface area of the cylinder to determine the amount of material required for its construction. However, she was unsure of the exact formula and the steps involved in the calculation. Can you help Emily by writing a C program that assists her in calculating the surface area of a cylinder?(Surface Area = $2\pi r^2 + 2\pi rh$)

10. Hemanthisanarchitectwhowantstodesignagardenwithabeautifulpolygon-shaped fountain at its center. He needs to calculate the area of the polygon so that he can determine the appropriate size for the fountain. Help Hemanth by writing a C program that calculates the area of a regular polygon when given the number of sides and the length of each side.

Area=(num Sides* side Length*side Length)/ (4 *tan(π/ num Sides))

11. Prathima loves ice cream cones and wants to decorate the surface of her favorite ice cream cone with colorful stickers. To know how many stickers she needs, she wants to calculate the surface area of the cone. Help Prathima by writing a C program that calculates the surface area of a cone when given the radius of the base and the slant height, (SurfaceArea=π*r*(r+l))

12. **RIGHT FOOT FORWARD:** A short man takes three steps to a tall man's two steps. They both start out on the left foot. How many steps do they have to take before they are both stepping out on the right foot together?

Week: 4DataTypes, Console I/O, Operators Storage Classes

PREDICTTHEOUTPUT:

1.
```
#include<stdio.h>
int main()
{
inta=5;floatb=3.5;
intresult=a+b;printf("%d\n",result);return0;
}
```
Output:

2.
```
#include<stdio.h>
intmain(){inta=10;intb=20;
intresult=a*b/4%3;printf("%d\n",result);return0;
}
```
Output:

3.
```
#include<stdio.h>
intmain(){inta=15;intb= 10;
intresult=(a>b)&&(b!=0);printf("%d\n", result); return0;
}
```
Output:

4.
```
#include<stdio.h>
intmain(){inta=10;intb= 5;
intresult=(a>b)||(a==10);printf("%d\n",result);
return0;
}
```
Output:

5.
```
#include<stdio.h>
int main(){
int x=3,y=2;
intresult=x*y-y/x%y;printf("%d\n", result);return 0;
}
```
Output:

6.
```
#include<stdio.h>
intmain(){intx=5;
 int *ptr1 = &x;
int**ptr2=&ptr1;printf("%d\n",**ptr2);return0;
}
```
Output:

7.
```
#include<stdio.h>
int main(){intx=5;
int*ptr1,*ptr2;ptr1 =&x;
ptr2=ptr1;printf("%d\n",*ptr2);return0;
}
```
Output:

8.
```
#include<stdio.h>
int main(){intx=51;
int*ptr=&x; printf("%d\n",*ptr);x=15;
 printf("%d\n",*ptr); return 0;
}
```
Output:

9.
```
#include<stdio.h>
intmain(){float*ptr;
printf("Sizeofptr:%lubytes\n",sizeof(ptr));return0;
}
```
Output:

10.
```
#include<stdio.h>
intmain(){doublearr[5];
printf("Sizeofarr:%lubytes\n",sizeof(arr));return0;
}
```
Output:

11.
```
#include<stdio.h>
intmain(){
intx=10;if(x>5){
printf("Hello\n");
}
printf("Goodbye\n");return0;
}
```
Output:

Skill Oriented Exercise

Find the syntax error, logical errors if any in the following code snippet:

1.
```
#include<stdio.h>
int main(){
printf("Hello,KLUFamily!\n")return0;
}
```
Errors:

2.
```
#Include<stdio.h>
intmain(){
int x = 5, y= 0;
intresult=x/y;printf("Theresultis:%d\n",result);
return0;
}
```
Errors:

3.
```
#include<stdio.h>
intmain(){intx=5;int*ptr;
*ptr=&x;printf("%d\n",*ptr);return0;
}
```
Errors:

4.
```
#include<stdio.h>
intmain(){intx=-5;if(x){
printf("xisnotzero\n");
}else
{
printf("xiszero\n";)
```
Errors:

```
}
return0;
}
```

Week5:If else and Ternary Operator

1. Chef and Chefina are playing with dice. In one turn, both of them roll their dice at once. They consider a turn to be good if the sum of the numbers on their dice is greater than
   6. Given that in a particular turn Chefand Chefinagot Xand Yon their respective dice, find whether the turn was good.

2. Chef has been working hard to compete in MasterChef. He is ranked *X* out of all contestants. However, only 10 contestants would be selected for the finals. Check whether Chef made it to the top 10 or not?

3. Apple considers any I Phone with a battery health of 80% or above, to be in *optimal* condition. Given that your iPhone has *X*% battery health, find whether it is in *optimal* condition.

4. In a classic chase, Tom is running after Jerry as Jerry has eaten Tom's favorite food. Jerry isrunningataspeedof*X*metrespersecondwhileTomischasinghimataspeedof *Y* metres per second. Determine whether Tom will be able to catch Jerry.
   Note that initially Jerry is not at the same position as Tom.

5. Chef has started studying for the upcoming test. The textbookhas *N*pages in total.Chef wants to read at most *X* pages a day for *Y* days. Find out whether it is possible for Chef to complete the whole book.

6. Chefhasfinallygotthechanceofhislifetimetodriveinthe*F*1tournament.But,thereis oneproblem.Chefdidnotknowaboutthe107%ruleandnowheisworriedwhetherhe will be allowed to race in the main event or not.
   Giventhefastestfinishtimeas*X*secondsandChef'sfinishtimeas*Y*seconds, determine whether Chef will be allowed to race in the main event or not.
   Note that, Chef will only be allowed to race if his finish time is within **107%** of the fastest finish time.

Skill Oriented Exercise

7. Chefwantstohostapartywithatotalof*N*people.However,thepartyhallhasa capacity of *X* people. Find whether Chef can host the party.

8. Chef has to attend an exam that starts in *X* minutes, but of course, watching shows takes priority.EveryepisodeoftheshowthatChefiswatching,is24minutes long. Ifhe starts watching a new episode now, will he finish watching it **strictly before** the exam starts?

9. Chef has to travel to another place. For this, hecanavail anyoneoftwocab services.
   - The first cab service charges *X*rupees.
   - The second cab service charges *Y*rupees.

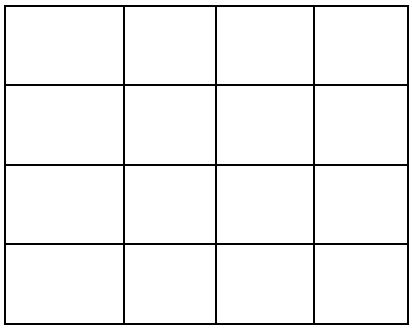
   Chef wants to spend the **minimum** amount of money. Which cab service should Chef take?

10. Chef categorizes an Instagram account as *spam*,if,the *following* count of the account is more than 10 times the count of *followers*.
Giventhe*following*and*follower*countofanaccountas*X*and*Y*respectively,find whether it is a *spam* account.

11. ChefiswatchingTV.ThecurrentvolumeoftheTVis*X*.Pressingthevolumeupbuttonofthe TVremoteincreasesthevolumeby11whilepressingthevolumedownbuttondecreasesthe volumeby11.Chefwantstochangethevolumefrom*X*to*Y*.Findtheminimumnumberof button presses required to do so.

12. Cities on a map are connected by a number of roads. The number of roads between each cityisinanarrayandcity0isthestartinglocation.Thenumberofroadsfromcity0tocity1 isthefirstvalueinthearray,fromcity1tocity2inthesecond,andsoon.Howmanypaths aretherefromcity0tothelastcityinthelist,modulo1234567?

13. Square within Square
WriteaCprogramtodeterminethenumberofsquaresinthegivenillustration below



Week6: Loops, Increment and Decrement operator

1. Sum of digits of five digit number.
Given the five digit number print the sum of its digits.

Input
10564
Output
16

2. Write a program that takes two integers as input, start and end. The program should use for loop to iterate from start to end (both inclusive).

3. WriteaCprogramthattakestheinputformultipletestcases.Foreachtestcase,the program should receive two integers, E and K, representing the energyat the lowest trophiclevelandtheenergyreductionfactor,respectively.Theprogramshouldcalculate and output the maximum length of the food chain for each test case.

4. Write a program in C that takes an integer, n, as input, representing the number of multiplicationtablestobegenerated.Theprogramshouldoutputthemultiplicationtable for each number from 1 to n, up to a multiple of 10.

5. Alice,Bob,andCharliehavedifferentpreferencesfornumbers.Alicelikesnumbersthat are even and multiples of 7, while Bob prefers numbers that are odd and multiples of 9.

They have found a number, A, and the task is to determine who takes it home.
Write a program that takes an integer, A, as input and outputs the person who takes the number home based on their preferences. If A is an even multiple of 7, Alice takes it home. If A is an odd multiple of 9, Bob takes it home. If neither Alice nor Bob likes the number, Charlie takes it home.

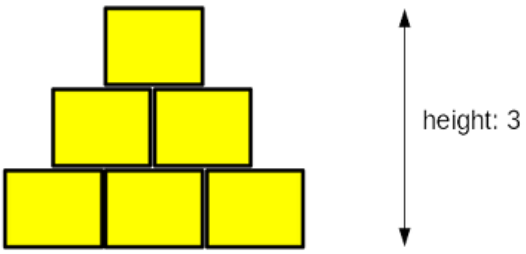
6. Chef owns a car that can run 15 kilometers using 1 litre of petrol. He wants to attend a programmingcampatDAIICT,whichisadistanceofYkilometersfromhishouse.Chef currently has X litres of petrol in his car. The task is to determine whether Chef can attend the event at DAIICT and return to his home with the given amount of petrol.Write a program that takes two integers, X and Y, as input and outputs whether Chef can complete the round trip with the available petrol.

Skill Oriented Exercise

7. Listen to this story: a boy and his father, a computer programmer, are playing with wooden blocks. They are building a pyramid. Their pyramid is a bit weird, as it is actually a pyramid- shaped wall - it's flat. The pyramid is stacked according to one simple principle: each lower layer contains one block more than the layer above. The figure illustrates the rule used by the builders:

Note: the height is measured by the number of fully completed layers –if the builders don't have a sufficient number of blocks and cannot complete the next layer, they finish their work immediately.

Sample input: 6
Expected output: The height of the pyramid: 3 Sample input:1000



blocks: 6

Expected output: The height of the pyramid:44

8. **The Special Number**
There is a number whose double is greater than its half by45.Can you find this number?

9. Chef is a student at a university, and the university has a requirement that students must be present foratleast75%oftheworkingdaysinasemestertopass.Thesemesterhasa total of 120 working days. Chef has been taking a lot of holidays and is worried about meeting the attendance requirement. The information about the days Chef has attended or been absent is given as a sequence of N bits: B1, B2,..., BN. If Bi =0, it means Chef was

absent on the ith day, and if Bi=1,itmeans Chef was present on that day. The task is to determine if Chef can pass the attendance requirement by the end of the semester. Write a program that takes an integer N as input, followed by a sequence of Nbits, and out puts whether Chef can hope to pass the attendance requirement or not.

10. There are N piles where the i^{th} pile consists of Ai stones. Chef and Chef in a are playing a game taking alternate turns with Chef starting first. In his/her turn, a player can choose any non-empty pile and remove exactly 1 stone from it. The game ends when exactly1 pile becomes empty. The player who made the last move wins. Determine the winner if both players play optimally.

11. Write the c program for following pattern

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

12. Write a program to obtain a number N and increment itsvalueby1ift he number is divisible by4 otherwise decrement its value by 1.

Week7: Arrays

1. Given a large integer represented as an integer array digits, where each digits[i]is the i-th digit of the integer, ordered from most significant to least significant in left-to-right order (without any leading zeros), implement a program to increment the large integer by one and return the resulting array of digits.

2. Chef has set a target to solve at least 10 problems every week for a duration of 4 weeks. The input consists of four integers representing the number of problems Chef solved in each week (P1, P2, P3, and P4). The task is to determine the number of weeks in which Chefmet his target. The output should be a single integer indicating the count of weeks where Chef solved at least 10 problems.
3. Code Chef recently revamped its practice page to make it easier for users to identify the next problems they should solve by introducing some new features: Recent Contest Problems - contains only problems from the last 2 contests Separate Un-Attempted, Attempted, and All tabs.

Problem Difficulty Rating-there commended drop down menu has various difficulty ranges so that you canattempttheproblemsmostsuitedtoyourexperiencePopularTopicsandTags. Like most users, Chef didn't know that he could add problems to a personal to-do list by clickingonthemagic'+'symbolonthetop-rightofeachproblempage.Butoncehefound out about it ,he went crazy and added loads of problems to his to-do list without looking at their difficulty rating.

Chef is a beginner and should ideally try and solve only problems with difficulty rating strictly lessthan1000.GivenalistofdifficultyratingsforproblemsintheChef'sto-dolist,please helphimidentifyhowmanyofthoseproblemsChefshouldremovefromhisto-dolist,sothat he is only left with problems of difficultyratinglessthan1000.

4. You are given an array price where prices[i] is the price of a given stock on the ith day. You want to maximize your profit by choosing a single day to buy one stock and

choosing a different day in the future to sell that stock. Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

5. Given a non-empty array of integer's nums, every element appears twice except for one. Find that single one. You must implement a solution with a linear runtime complexity and use only constant extra space.

6. Given an array nums of size n, return the majority element. The majority element is the element that appears more than ⌊n / 2⌋ times. You may assume that the majority element always exists in the array.

7. Write a C program to calculate the factorial of small positive integers. The input consists of an integer 't' representing the number of test cases, followed by 't' lines containing a single integer 'n' (1 <= n <= 100) for each test case. The output should display the factorial of 'n' on a separate line for each input value of 'n'.

8. (Puzzle)On the Way to Market One morning.
I was on my way to the marketandmetamanwhohad4wives.Eachofthewiveshad4 bags ,containing 4 dogsandeachdoghad4puppies.Taking all things into consideration how many were going to the market?

9. Vasyalikesthenumber239.Therefore,he considers a number pretty if its last digit 2,3 or 9.Vasya wants to watch the numbers between Land R(both inclusive),so he asked you to determine how many pretty numbers are in this range. Can you help him?

10. You are participating in a contest which has 11 problems (numbered 1 through 11). The first eight problems (i.e. problems 1,2,…,8) are scorable, while the last three problems (9,10 and 11) are non-scorable — this means that any submissions you make on any of these problems do not affect your total score.
Your totals core is the sum of your best scores for all scorable problems. That is, for each scorable problem, you look at the score so fall submissions you made on that problem and take the maximum of these scores(or0if you didn't make any submissions on that problem); the total score is the sum of the maximum scores you took. You know the results of all submissions you made. Calculate your total score.

Skill Oriented Exercise

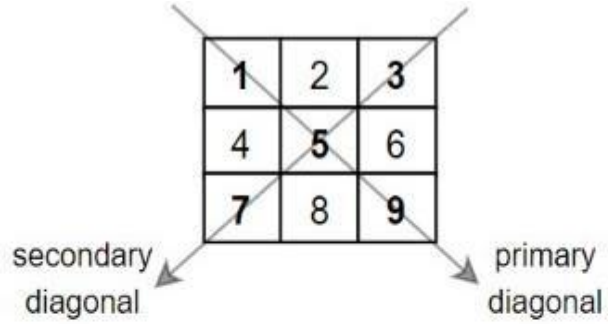
11. WriteaCprogramtohelpJoeandLillymultiplytwomatrices,AandB.Theprogram should take input for multiple test cases. For each test case, the program should read the dimensions and values of matrices A and B. If the multiplication is possible, the program should print the output matrix values. If the multiplication is not possible, the program should print "IMPOSSIBLE".

12. You are given an m x n integer matrix with the following two properties:
Each row is sorted in non-decreasing order. The first integer of each row is greater than the last integer of the previous row. Given an integer target, return true if target is in matrix or false other wise.

1	3	5	7
10	11	16	20
23	30	34	60

Input :matrix=[[1,3,5,7],[10,11,16,20],[23,30,34,60]],target=3
Output: true

13. You are given an m x n integer grid accounts where accounts[i][j] is the amount of money the ith customer has in the jth bank. Return the wealth that the richest customer has A customer's wealth is the amount of money they have in all their bank accounts. The richest customer is the customer that has the maximum wealth.
14. Given a square matrix mat, return the sum of the matrix diagonals. Only include the sum of all the elements on the primary diagonal and all the elements on the secondary diagonal that are not part of the primary diagonal.



Input: mat = [[**1**,2,**3**],
[4,**5**,6],
[**7**,8,**9**]]
Output: 25
Explanation: Diagonals sum: 1 + 5 + 9 + 3 + 7 = 25
Notice that element mat[1][1] = 5 is counted only once.

15. Writeaprogramtoperformmatrixmultiplication.IfMultiplicationcannotbedonefora given matrices then print "NOT POSSIBLE"

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix}$$

Multiplication of two matrixes:

$$A * B = \begin{pmatrix} 1*5 + 2*8 & 1*6 + 2*9 & 1*7 + 2*10 \\ 3*5 + 4*8 & 3*6 + 4*9 & 3*7 + 4*10 \end{pmatrix}$$

$$A * B = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

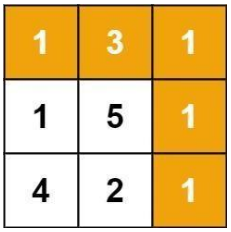
Input:
1) Read the row & column sizeofmatrix1
2) Readthematrix1
3) Read the row & column sizeofmatrix2
4) Readthematrix2

Output:
Resultant Matrix.

Sample Input	Sample Output
22	710
12	1522
34	
22	
12	
34	

16. Given two sorted arrays num s1and num s2ofsizemandnrespectively,return the median of the two sorted arrays.
Input: nums1 = [1,3], nums2 = [2] Output: 2.00000
Explanation: merged array=[1,2,3] and median is 2.

17. Given a m x n grid filled with non-negative numbers, find a path from top left to bottom right, which minimizes the sum of all numbers along its path.
Note: You can only move either down or right at any point in time.



Input: grid=[[1,3,1],[1,5,1],[4,2,1]]
Output:7
Explanation: Because the path1→3→ 1→1→1 minimizes the um.

18. Given an array nums of size n, return the majority element. The majority element is the element that appears more than ⌊n/2⌋ times. You may assume that the majority element always exists in the array.
Input: nums=[3,2,3]Output:3

19. Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

Input: nums=[1,3,5,6],target=5Output:2

20. Givenamxnmatrixgridwhichissortedinnon-increasingorderbothrow-wise and column-wise, return the number of negative numbers in grid.

```
Input: grid = [[4,3,2,-1],[3,2,1,-1],[1,1,-1,-2],[-1,-1,-2,-3]]
Output: 8
Explanation: There are 8 negatives number in the matrix.
```

Example 2:

```
Input: grid = [[3,2],[1,0]]
Output: 0
```

Week8: String Handling

1. Louise joined a social networking site to stay in touch with her friends. The signup page required her to input a *name* and a *password*. However, the password must be *strong*. The website considers a password to be *strong* if it satisfies the following criteria:
   - Its length is at least 6.
   - It contains at least one digit.
   - It contains at least one lower case English character.
   - It contains at least one upper case English character.
   - It contains at least one special character. The special characters are: @#$%^&*()-+!

2. As pace explorer's ship crashed on Mars! They send a series of SOS messages to Earth for help.

Letters in some of the SOS messages are altered by cosmic radiation during transmission. Given the signal received by Earth as a string, determine how many letters of the SOS message have been changed by radiation.
s='SOSTOT'

Example

The original message was SOSSOS. Two of the message's characters were changed in transit.

Function Description

CompletethemarsExplorationfunctionintheeditorbelow.marsExplorationhasthefollowing parameter(s):
strings: the string as received on Earth

Returns

int: the number of letters changed during transmission

3. ChefhasastringSwithhim.Chefishappyifthestringcontainsacontiguoussubstring of length strictly greater than 2 in which all its characters are vowels.
   Determine whether Chef is happy or not.
   Note that, in english alphabet, vowels are a, e, i, o, and u.
4. Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.
5. Givenastringsconsistingofwordsandspaces,returnthelengthofthelast word in the string. A word is a maximal substring consisting of non-space characters only.

6. Given a string S, reverse only all the vowels in the string and return it. The vowelsare'a','e','i','o',and'u',andtheycanappearinbothlowerandupper cases, more than once.
7. You have been given a String S. You need to find and print whether this string is a palindrome or not. If yes, print "YES" (without quotes), else print "NO" (without quotes).

Skill Oriented Exercise

8. Jeff, Chef's younger brother, is learning to read and knows a subset of the Latin alphabet. Chef gave Jeff a book with N words to practice. Jeff can only read wordsthatconsistofthelettersheknows.Thetaskistodeterminewhichwords Jeff can read based on the given letters and output "Yes" or "No" accordingly.

9. Timur loves code forces. That's why he has a string Shaving length 10made containing only lower case Latin letters. Timur wants to know how many indices string s differs from the string "codeforces".

Forexamplestrings="coolforsez"differsfrom"codeforces"in4indices,showninbold.

Up the Ladder

A man wants to reach window which is 40ft above from the ground. And the distance between the foot of the ladder and wall is 9 feet. How long should the ladder be?

10. Given two strings s and t, return true if t is an anagram of s, and false otherwise. An Anagramisawordorphraseformedbyrearrangingthelettersofadifferentword or phrase, typically using all the original letters exactly once. Given strings, find the first non-repeating character in it and return its index. If it does not exist, return -1.

11. A robot starts at the origin (0, 0) on a 2D plane. It is given a sequence of moves represented by the string "moves". Each move is represented by 'R' (right), 'L' (left), 'U' (up), or 'D' (down). The task is to determine if the robot returns to the origin after completing all the moves. The robot's direction is irrelevant, and all moves have the same magnitude. Return true if the robot ends up at the origin, and false otherwise.

12. Chandu is a bad student. Once his teacher asked him to print the reverse of a given string. He took three hours to solve it. The teacher got agitated at Chandu and asked you the same question. Can you solve it?

13. There is a string s of lowercase English letters that is repeated infinitely many times. Given an integer n find and print the number of letter a's in the first n letters ofthe infinite string.

Week9: Recursion

1. Given an integer n, return true if it is a power of three. Otherwise, return false. An integer n is a power of three, if there exists an integer x such that n == 3^x.

2. You are climbing a stair case. It takes n steps to reach the top. Each time you can either climb1or 2steps. In how many distinct ways can you climb to the top?

3. Given an integer n, return true if it is a power of four. Otherwise, return false. An integer n is a power of four, if there exists an integer x such that n == 4^x.

4. You are given an integer N. You need to print N! –the factorial of N.

Input The first line of the input contains a single integer *T* denoting the number of test cases. The description of *T* test cases follows. The first and only line of each test case contains a single integer *N*.

Output For each test case print a single line containing a single integer *N*!

5. The Fibonacci numbers, commonly denoted F(n) form a sequence, called the Fibonacci sequence, such that each number is the sum of the two precedingones,startingfrom0and 1

6. Kristenlovesplayingwithandcomparingnumbers.Shethinksthatifshetakestwodifferent positive numbers, the one whose digits sum to a larger number is *better* than the other. If the sum of digits is equal for both numbers, then she thinks the smaller number is *better*. For example, Kristen thinks that 13is better than 31and that12is better than 11 .Given an integer, n ,can you find the divisor of n that Kristin will consider to be the best?

7. A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. A divisor of an integer x is an integer that can divide x evenly. Given a n integer n, return true if n is a perfect number, other wise return false.

8. Given an integer num, repeatedly add ll its digits until the result has only one digit, and return it.

9. **Something for Profit:** A friend of mine bought a used pressure cooker for Rs. 60. She somehow did not find it useful and so when a friend of hers offered her Rs. 70 she sold it to her. However, she felt bad after selling it and decided to buy it back from her friend' by offering her Rs. 80. After having bought it once again she felt that she did not really need the cooker. So, she sold it at the auction forRs.90.Howmuchprofitdidshemake? Did she a tall make any profit?

10. Given a signed 32-bit integer x, return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range [-2^{31}, 2^{31} - 1], then return 0.

11. Given an integer array num s, move all 0's to the end of it while maintaining the relative order of the non-zero elements.

12. Writeafunctionthattakesthebinaryrepresentationofanunsignedinteger and returns the number of '1' bits it has (also known as the Hamming weight).

13. Martha is interviewing at Subway. One of the rounds of the interview requires her to cut a bread of size l X b into smaller identical pieces such that each piece is a square having maximum possible side length with no leftover piece of bread.

14. Given N two-dimensional points in space, determine whether they lie on some vertical or horizontal line.

If yes, print YES; otherwise, print NO.

Skill Oriented Exercise

PREDICTTHEOUTPUT

```
#include<stdio.h>voidfoo(intn){
if(n>0){printf("%d",n);foo(n-1);printf("%d",n);
}
}
intmain(){
foo(3);return0;
}
```

1. **What is the output of the above program?**
   - 321123
   - 321
   - 123
   - 123321

```
#include<stdio.h>intbar(intn){ if (n
<= 0) { return 0;
}else{
returnn+bar(n-2);
}
}
int main(){
intresult=bar(7); printf("%d",result);return0;
}
```

2. **What is the output of the above program?**
   - 20
   - 16
   - 14
   - 12

```
#include<stdio.h>voidbaz(intn){ if (n >
0) {
baz(n/2);printf("%d",n%2);
}
}
intmain(){
baz(10);return0;
}
```

3. **What is the output of the above program?**
   - 1010
   - 0101
   - 00101
   - 1101

```
#include<stdio.h> intfactorial(intn){ if (n
```

```
== 0) { return 1;
}else{
returnn*factorial(n-1);
}
intmain(){
intresult=factorial(5); printf("%d",result);return0;
}
```

4. What is the output of the above program?

- 120
- 24
- 25
- 20

```
#include<stdio.h>
intpower(intbase,intexponent){if(exponent==0){return1;
}else{
returnbase*power(base,exponent-1);
}
}

intmain(){
intresult=power(2,4);printf("%d",result); return0;
}
```

5. What is the output of the above program?

- 16
- 8
- 32
- 64

Week10: Structures

1. You are building a pay roll system for accompany with multiple departments. Designa programusingstructuresthatstoresemployeedetailssuchasname,employeeID,and salary. Implement an array of structures to store employee records for each department. Calculate the total salary expenditure for each department and display it. Additionally, identify the department with the highest salary expenditure and acknowledge it as the top-performing department

2. VGP logistics is a premium Cargo service for Sending/receiving parcels from Vijayawada to Singapore. You are appointed as Manager in delivery department and need to maintain the arrival and delivery of the consignments sent/received. Create a structure consignment with the following Members Consignment_id, name, from, to, DOS (Date of Shipment), net weight, Address.
   For Example:
   Consignment_id:1008Name:HaierSteamerFrom:Vijayawada To: Singapore
   DOS(Date of Shipment):30-may-2023net_weight:28.8kg Address: Mint Street Chennai

3. You have been assigned the task of developing a student grading system for a prestigious college. Design a program using structures that stores student details, such as name, roll number, and marks in various subjects. Implement an array of structures to store multiple

student records and calculate the overall percentage for each student. Additionally, provide a functionality to generate a grade for each student based on their percentage and display it alongside their record.

4. You are developing a soldier management system for an army unit. Each soldier's record consists of the following information: name, rank, and years of service. Implement an array of structures to store the records of multiple soldiers. Write a C program to calculate and display the average years of service for all soldiers in the unit.

5. You are working on a ship management system for a naval fleet. Each ship's record contains the following details: name, type (e.g., aircraft carrier, destroyer), and year of commissioning. Implement an array of structures to store the ship records. Write a C program to search for a specific type of ship within the fleet and display the names of all ships belonging to that type.

6. You have been assigned the task of developing a pilot roster system for an air force squadron. Each pilot's record includes the following information: name, rank, and flight hours. Implement an array of structures to store the pilot records. Write a C program to find and display the pilot with the highest number of flight hours in the squadron.

7. You are working on a reservation system for a luxurious hotel. Create a program using structures that stores guest details, including name, room number, and check-in date. Implement an array of structures to store multiple guest records and allow the hotel staff to search for guests by either their room number or name. Provide an additional feature that calculates the duration of each guest's stay and generates the total revenue earned by the hotel.

Skill Oriented Exercise

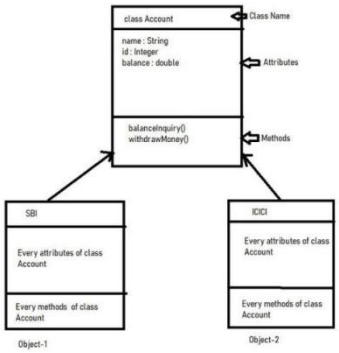
8. You have been tasked with developing a library management system for a renowned library. Create a program using structures that stores book information, including title, author, and publication year. Implement an array of structures to store multiple book records and allow the librarian to search for books by either title or author's name. Enhance the system by enabling the librarian to borrow and return books, updating the book status accordingly.

9. You have been assigned the task of creating a customer billing system using an array of structures. Each structure should store the customer's name, account number, and total amount due. Implement functions to add customer records, display all records, and find the customer with the highest amount due.

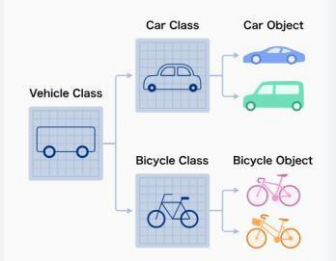
10. You are working on a car inventory management system using an array of structures. Each structure should hold the details of a car, including the make, model, and year of manufacture. Implement functions to add car records, display all records, and find the newest car in the inventory.

11. You are developing a product inventory management system for a retail store. Each product has a unique identifier, name, price, and quantity in stock. Implement an array of structures to store the product records. Create functions to add new products, update product details, display all products, and search for products based on their identifier or name.

Week11: Classand Objects:



1. Create a class and the object code for the above scenario.



2. Develop code for Class and Object.

Week12: Data Abstraction

Developing a banking application that handles various types of accounts such as savings, checking, and loans.

- **Abstraction Use:** Create abstract classes or interfaces like Account with common methods (e.g., deposit(), withdraw(), get Balance()). Concrete classes (e.g., Savings Account, Checking Account) implement these methods. The user interacts with Account objects without needing to know the specific type of account or its internal workings.

Inheritance

Developing a graphical user interface (GUI)library with various types of buttons.

- **Inheritance Use:** Create a base class Button with common properties like label, size, and methods like click(). Derive subclasses like Image Button, Toggle Button, and RadioButtonthatinheritfromButtonandaddspecificpropertiesormethodsuniqueto each type of button.

Building a ne-commerce plat form with different types of products.

- **Inheritance Use:** Define a base class Product with common attributes like name,

price, and methods like apply Discount(). Create subclasses like Electronics, Clothing, and Books, each adding specific attributes (e.g., Electronics might have warrantyPeriod, Clothing might have size and color).

Skill Oriented Exercise

Building educational software that manages different types of learning resources.

- **Inheritance Use:** Establish a base class Learning Resource with common properties like title, subject, and methods like display(). Derive subclasses such as Book, Video Lecture, and Quiz, each adding specific attributes and methods (e.g., Quizmight have questions and a method evaluate()).

Designing a system to classify and manage information about different animals.

- **Inheritance Use:** Define a base class Animal with common attributes like name, habitat, and methods like eat(), sleep(). Create subclasses such as Mammal, Bird, and Reptile, each with specific characteristics and methods(e.g., Bird might have methods fly()).

Week13: Polymorphism

An application that can draw various shapes such as circles, rectangles, and triangles.

- **Polymorphism Use:** Define an abstract class Shape with a method draw(). Implement subclasses Circle, Rectangle, and Triangle, each providing its specific draw() implementation. The application can then handle any shape object through the Shape interface

A program that simulates sounds of different animals.

- **Polymorphism Use:** Create a base class Animal with an abstract method make Sound(). Implement subclasses Dog, Cat, and Cow that override make Sound(). The simulator can then invoke make Sound() on any animal object

Managing different types of transportation such as cars, buses, and bicycles.

Skill Oriented Exercise

- **Polymorphism Use:** Define a base class Vehicle with an abstract method move(). Implement subclasses Car, Bus, and Bicycle, each with its own implementation of move(). The system can then manage different vehicles uniformly

A system sending notifications via email, SMS, and push notifications.

- **Polymorphism Use:** Define an abstract class Notification with a method send(). Implement subclasses Email Notification, SMS Notification, and Push Notification, each with its own send() method. The system can send notifications through any medium using the same interface

Week14: Virtual functions

A document edit or that supports different types of documents such as text documents, spreadsheets, and presentations.

- **Virtual Function Use:** Define a base class Document with a virtual function save(). Subclasses Text Document, Spreadsheet, and Presentation override save() to handle specific saving mechanisms.

Anaudioprocessinglibrarythathandlesvariousaudioeffectssuchasreverb,echo,and distortion.

Skill Oriented Exercise

- **Virtual Function Use:** Define a base class Audio Effect with a virtual function apply().SubclassesReverbEffect,EchoEffect,andDistortionEffectoverrideapply( ) to implement specific effects.

An AI strategy game that involves different types of game characters such as warriors, mages, and archers.

- **Virtual Function Use:** Define a base class Character with a virtual function attack(). Subclasses Warrior, Mage, and Archer override attack() to provide specific attack behaviors.

Week15: Exception handling

1) File Operations Scenario: An application needs to read data from a file. Exception Handling Use: Implement code to handle scenarios where the file might not exist, the application lacks permissions, or the file is corrupted. Use try-catch blocks to manage these exceptions:

2) E-Commerce Checkout Process

Scenario:Ane-commerceapplicationprocessesuserordersduringcheckout.

- Exception Handling Use: Handle errors such as invalid payment details, out-of-stock items, or delivery address issues.

Skill Oriented Exercise

3)An application processes images for various operations like resizing, filtering, and saving.

- Exception Handling Use: Handle errors such as unsupported file formats, corrupted files, or out-of-memory issues.

TEXTBOOKS:

1. ForouzanB.A&RichardF.Gilberg,A Structured ProgrammingApproachusingC,3rd Edition(2013), Cengage Learning.
2. JeriR.HanlyandElliotB.Koffman,ProblemsolvingandProgramDesigninC7th Edition, Pearson
3. ANSI and Turbo C++by Ashoke N.Kamthane, Pearson Education

REFERENCES:

1. BrianW.KernighanandDennisM.Ritchie,TheCProgrammingLanguage,PrenticeHall of India
2. E.Balagurusamy,ComputerfundamentalsandC,2ndEdition,McGraw-Hill
3. YashavantKanetkar,LetUsC,18thEdition, BPB
4. E.Balagurusamy,ObjectOrientedProgrammingusingC++,2ndEdition,McGraw-Hill

24X0073: ENGLISH LANGUAGE AND COMMUNICATION SKILLS LABORATORY

B.Tech. I Year. I 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 the students with the use of English in everyday situations both in formal and informal contexts.

Course Objective

- ✓ To facilitate computer-assisted multi-media instruction enablingindividualized and independent language learning
- ✓ To sensitize the students to the nuances of English speech sounds, wordaccent, intonation 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 theimpact of dialects.
- ✓ To train students to use language appropriately for public speaking,group discussions and interviews

Course Outcomes: Students will be able to:

- ✓ Understand the nuances of the English language through audio-visual experienceand group activities
- ✓ Neutralize their accent for intelligibility
- ✓ Speak with clarity and confidence which in turn enhances their employability skills

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 to develop their listening skills so that they may appreciate therole 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 to recognize and use the right intonation in sentences.

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

Speaking Skills:

Objectives

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

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

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

Exercise – ICALL 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- *TestingExercises*

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 –

IICALL

Lab:

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

Practice: Basic Rules of Word Accent - Stress Shift - Weak Forms and Strong Forms- Stresspattern 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 –MakingRequests and Seeking Permissions - Telephone Etiquette.

Exercise -

IIICALL

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 – IVCALL Lab:

Understand: Listening for General Details.

Practice: Listening Comprehension Tests - *Testing Exercises*

ICS Lab:

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

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

Exercise –

VCALL

Lab:

Understand: Listening for Specific Details.

Practice: Listening Comprehension Tests -*Testing Exercises*

ICS Lab:

Understand: Group Discussion

Practice: Group Discussion

Minimum Requirement of infrastructural facilities for ELCS Lab:

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

The Computer Assisted Language Learning Lab has to accommodate 40 students with 40 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 40 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, a T.V. or LCD, a digital stereo –audio & video system and camcorder etc.

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 thecontents of 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) CambridgeUniversity Press.
- English Vocabulary in Use (Elementary, Intermediate, Advanced) CambridgeUniversity Press.
- TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS).
- Digital All
- Orell Digital Language Lab (Licensed Version)

REFERENCE BOOKS:

1. (2022). *English Language Communication Skills – Lab Manual cum Workbook.* Cengage Learning India Pvt. Ltd.
2. Shobha, KN & Rayen, J. Lourdes. (2019). *Communicative English – A workbook.* Cambridge University Press
3. Kumar, Sanjay & Lata, Pushp. (2019). *Communication Skills: A Workbook.* OxfordUniversity Press
4. Board of Editors. (2016). *ELCS Lab Manual: A Workbook for CALL and ICS Lab Activities.* Orient Black Swan Pvt. Ltd.
5. Mishra, Veerendra et al. (2020). *English Language Skills: A Practical Approach.* Cambridge University Press.

2410596: WEB APPLICATION DEVELOPMENT

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

COURSE OBJECTIVES: The students will try to learn

- HTML tags
- CSS
- Development of static web site
- Concepts of Java script
- Development of dynamic web site

0 0 2 1

COURSE OUTCOMES: After successful completion of the course, students should be able to

- Learn HTML tags and CSS
- Develop static web pages using HTML
- Use CSS in web pages
- Understand basic concepts of Javascript
- Develop dynamic web pages

PART-A

1. Write a HTML program to create a webpage about the different art forms of India, with appropriate title on the title bar. Use different heading tags for the headings, and list them using ordered list.
2. Write a HTML program to create sections in the document using appropriate tags and apply different color as background to them. Use internal hyperlinks to move to differentpoints within the page.
3. Write a HTML program to insert a picture on the webpage, giving description for the picture in a paragraph. Use properties of height, width, hspace, vspace and align, with different values.
4. Write a HTML Program, to create a profile of 2 pages, the First page containing the applicant's picture with personal details using unordered lists, and the second containingEducational details using tables. Use hyperlinks to move to the next page.
5. Using Frames create an Indian Flag and insert the image of chakra in the center.
6. Create a frame like structure based on the given diagram, such that When the first link is clicked, the contents of the first frame is filled with the corresponding information and when the second link is clicked the second frame is filled.

<table>
<tr><td rowspan="2">Networks
Simulation</td><td>Networks……..</td></tr>
<tr><td>Simulation</td></tr>
</table>

7. Write a program in HTML to demonstrate the concept of Image map, for India map.Map for areas rectangle, Circle and polygon.

PART-B

1. Write a program using Javascript to do the multiplication table for a number entered bythe user in the textbox.
2. Create a sparse array using the values entered by the user in the five textboxes, and useArray methods such as sort(),pop(),push(), reverse() and join().
3. Create a Math object and use methods ceil(),floor(), round() for rounding off thenumber, also use methods such as cos(), sin(),sqrt().
4. Write a Program using Javascript to print a bill for 5 items purchased by the user.
5. Write a program Using Date object, to display appropriate greeting message "Good Morning" or "Good Afternoon" or "Good Night", in an alert box with the user's name, after receiving the same in the prompt box.
6. To demonstrate the concept of styles, Write a program applying internal style for paragraph tag, and override the same by applying inline style. Also create an external CSS file applying styles for the headings.
7. Create a registration form for creating an email account, having the input type elements like checkbox, radio button, select option, text area and submit button, and validate the textboxes for verifying the password.
8. Create a web page using two image files, which switch between one another as the mouse pointer moves over the image. Use on Mouse Out and on Mouse Over event handlers.
9. Using filters apply opacity feature to blur the image and using Transition apply hover feature, so the image will be transparent again when the mouse pointer is placed on the image.

I–II

2420002: DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS
(CSE, CSD, CSM, ECE, EEE, MECH, CIVIL)

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

3 1 0 4

Course Overview:

This course plays a crucial role in engineering, serving as the foundation upon which engineers build and apply their knowledge to solve real world applications. It presents a systematic and comprehensive introduction to ordinary differential equations and vector calculus for engineering students .Mathematical concepts and various techniques are presented in a clear logical and concise manner. A linear differential equation is used to regulate the flow of electricity in various electrical circuits like LR, LCR and CR circuits. Vector calculus is extensively used in the description of electromagnetic fields, gravitational fields and fluid flow.

Pre-requisites: Mathematics courses of 10+2 year of study.

Course Objectives: The student will try to learn

- Methods of solving the differential equations of first order and first degree.
- Concept of higher order liner differential equations.
- Concept, properties of Laplace transforms, solving ordinary differential equations by 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.

Course outcomes: After successful completion of the course, students should be able to

CO1: Identify whether the given first order differential equation is exact or not.

CO2: Solve higher differential equation and apply the concept of differential equation to real world problems.

CO3: Usethe Laplace transforms techniques for solving ODE's.

CO4: Apply the Del operator to scalar and vector point functions.

CO5: Evaluate the line, surface and volume integrals and converting them from one to another.

UNIT-I: First Order ODE 10L

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, $e^{ax}V(x)$ and x V(x), method of variation of parameters, Equations reducible to linear ODE with constant coefficients: Legendre's equation, Cauchy-Euler equation.

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 (All without proof), Evaluation of integrals by Laplace transforms, Laplace transform of periodic functions, Inverse Laplace transform by different methods, convolution theorem (without proof). Applications: solving Initial value problems by Laplace Transform method.

UNIT-IV: Vector Differentiation 8 L

Vectorpointfunctionsandscalarpointfunctions,Gradient,DivergenceandCurl,Directionalderivatives, Vector Identities, Scalar potential functions, Solenoidal and Irrotational vectors.

UNIT-V:Vector Integration 10 L

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

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, 9thEdition, 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.

24X0009: ENGINEERING CHEMISTRY

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

3 1 0 4

Course overview:

1. **Importance of course-** It helps engineers understand the nature of different materials. It also helps engineers learn how to work with different types of matter without causing pollution or waste.

2. **Brief about course-** To impart knowledge on the fundamental concepts of chemistry Involved in application of several important engineering materials that are used in Industry/day to day life**.**

3. **Applications of course:** Engineering chemistry graduates use raw materials and chemicals to design, manufacture, and test new products, systems and machinery which are used in numerous industries.

Prerequisites: chemistry knowledge at Pre- University level

Course Objectives: The students will try to learn

1. Knowledge about desalination of brackish water and treatment of municipal water.
2. Fundamental aspects of battery chemistry, significance of corrosion its control to protect the structures.
3. Knowledge of polymers, conducting polymers, bio-degradable polymers and fiber reinforced plastics.
4. Basic concepts of petroleum and its products.
5. Knowledge about engineering materials like cement, smart materials and Lubricants.

Course Outcomes: After successful completion of the course, students should be able to

1. Apply softness of water by ion exchange process.
2. Analyze the various types Factors affecting of corrosion.
3. Understand the fundamental concepts of polymers
4. Analyze the various type of Gaseous Fuels.
5. Evaluate the smart materials and their Applications.

Course articulation matrix

PO/PSO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	4	2	1	-	-	1	-	-	-	-	1	1	-
CO2	2	4	3	4	2	1	1	1	-	-	-	1	1	-
CO3	1	2	1	2	1	2	1	-	-	-	-	1	1	-
CO4	1	4	4	5	1	2	1	-	-	-	-	1	1	-
CO5	1	4	4	5	1	2	1	-	-	-	-	1	1	-

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

Introduction to hardness of water – Estimation of hardness of water by complex

ometric 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.

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 Brackish water – Reverse osmosis.

UNIT – II Battery Chemistry & Corrosion [8]

Introduction-Classification of batteries-primary, secondary and reserve batteries with examples. Basic requirements for commercial batteries. Construction, working and applications of: Lithium ion battery, Applications of Li-ion battery to electrical vehicles. Fuel Cells- Differences between battery and a fuel cell.

Construction and applications of Hydrogen Oxygen 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. Surface coatings-Metallic coatings-Hot dipping-Galvanisation, Tinning

UNIT - III: Polymeric materials: [8]

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

Plastics: Definition and characteristics- thermoplastic and thermosetting plastics, Preparation, Properties and engineering applications of PVC ,Teflon, Fiber reinforced plastics (FRP).

Rubbers: Natural rubber and its vulcanization.

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

Conducting polymers: Characteristic, Classification and applications of conducting polymers.

Biodegradable polymers: Concept and advantages - 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,. Knocking – octane and cetane rating, synthetic petrol - Fischer-Tropsch's process; Gaseous fuels – composition and uses of natural gas, LPG and CNG.Bio-diesel-Transesterification-advantages.

UNIT - V: Engineering Materials: [8]

Cement: Portland cement, its composition, setting and hardening, special cements-white cement, waterproof cement, high alumina cement.

Smart materials and their applications:-

Classification-(piezoelectric materials-quartz, Shape memory material (SMA-Nitinol), Thermoresponsive materials, magneto rheological materials-Examples.

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.

TEXT BOOKS:

1. Engineering Chemistry by P.C. Jain and M. Jain, Dhanpatrai Publishing Company, 2010.
2. Engineering Chemistry by Rama Devi, and Rath, Cengage learning, Second edition 2022.
3. Textbook of Engineering Chemistry by Jaya Shree Anjireddy, Wiley Publications,2022.
4. A text book of Engineering Chemistry by M. Thirumala Chary, E. Laxminarayana and K. Shashikala, Pearson Publications, 2021.
5.

REFERENCE BOOKS:

4. Engineering Chemistry by Shikha Agarwal, Cambridge University Press, Delhi (2015)
5. Engineering Chemistry by Shashi Chawla, Dhanpatrai and Company (P) Ltd. Delhi(2011)

(24X0201) Principles of Electrical and Electronics Engineering
(Common to CSE,CSM and CSD)

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

COURSE OVERVIEW:

This Course provides the essential principles and theories Important in various aspects such as practical applications in daily life like household appliances, lighting systems, and personal electronics and industry use and technology.

It covers essential principles and concepts related to electrical and electronics systems promoting critical thinking, problem-solving, and analytical skills. It provides overview on basic definitions of electrical and electronics engineering, DC and AC circuits and theorems. It also gives knowledge about characteristics and applications of electronics devices.

Prerequisite: NIL

COURSE OBJECTIVES: The students will be able

- To analyze and solve electric circuits.
- To provide an understanding of basics in Electrical circuits and identify the types of electrical machines for a given application.
- To analyze the LT switchgear components
- To explain the characteristics of Electronics devices.

COURSE OUTCOMES: After successful completion of the course, students should be able to

- Analyze DC Electrical circuits to compute and measure the parameters.
- Analyze AC Electrical circuits with phasor representations
- Comprehend the working principles of Electrical Machines.
- Comprehend the components of LT Switchgear.
- Identify and test various characteristics of electronics devices.

MODULE –I

DC Circuits: Electrical circuit elements (R, L and C), voltage and current sources, KVL &KCL, analysis of simple circuits with dc excitation, Superposition theorem, Norton's Theorem and Thevenin's Theorem.

MODULE – II

AC 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, Three-phase balanced circuits, voltage and current relations in star and delta connections.

MODULE -III

Transformers: Construction and working principle of Single-phase transformer, equivalent circuit, losses in transformers and efficiency.

DC Machines: Construction and working principle of DC generators, EMF equation, working principle of DC motors and Torque equation.

MODULE -IV

Components of LT Switchgear: Switch Fuse Unit (SFU), MCB, ELCB, MCCB, Types of wires, cables and earthing.

Batteries: Types of batteries, important characteristics for batteries, elementary calculations for energy consumption, power factor improvement and battery backup.

MODULE -V

PN junction Diode: Volt-Ampere characteristics, applications, Static and dynamic resistances

Zener Diode: Volt-Ampere characteristics and it's applications.

Rectifiers: Half wave Rectifier, full wave rectifier, Bridge Rectifier-Ripple factor, efficiency and peak inverse voltage.

Text Books:

1. Basic Electrical Engineering - By M.S. Naidu and S. Kamakshaiah – TMH.
2. Basic Electrical Engineering –By T.K. Nagasarkar and M.S. Sukhija Oxford University Press.
3. Electronic Devices and Circuits- J. Millman, C. Halkias, Tata Mc-Graw Hill, Second Edition.
4. Integrated Electronics- Jacob Millman, C. Halkies, C.D.Parikh, Tata Mc-Graw Hill, 2009.

Reference Books:

1. Theory and Problems of Basic Electrical Engineering by D.P. Kothari& I.J. Nagrath PHI.
2. Principles of Electrical Engineering by V. K Mehta, S.Chand Publications.
3. Essentials of Electrical and Computer Engineering by David V. Kerns, JR. J. David Irwin Pearson.
4. Electronic Devices and Circuits-K. Satya Prasad, VGS Book Links.
5. Electronic Devices and Circuits - Salivahanan, Kumar, Vallavaraj, Tata Mc-Graw Hill, Second Edition
6. Electronic Devices and Circuits – Bell, Oxford.

24X0502: Essentials of Problem Solving Using Python

B.Tech. I Year II Sem.

L	T	P	C
3	0	0	3

Course Overview:

This course gives acquaintance to Python Programming and Graph Theory. The course deals with Python programming concepts and concepts in graph theory like properties of standard graphs, Eulerian graphs, Hamiltonian graphs, Chordal graphs, Distances in graphs, Planar graphs, graph connectivity and Colouring of graphs.

Graph theory is used in Network Topologies and Routing Algorithms, Algorithm Design, Social Network Design, Logistics.

Prerequisites:

- A course on "Problem solving using C and C++".

Course Objectives: The students will try to learn

- Basic building blocks of python
- Using of Functions and Modules
- Importance of Multithreading in problem solving
- The fundamental concepts of graph theory
- Graph coloring and traversal algorithms for solving real-world problems

Course Outcomes: After successful completion of the course, students should be able to

- Construct Python data structures programs using tuples sets and dictionaries
- Design Programs using Functions and Modules
- Implement Multithread concept in solving problems
- Understand graph terminology
- Build efficient graph routing algorithms for various optimization problems on graphs.

Module-I **[10]**

Python Basics: Python Objects, Operators, Python Numbers, Operators, Built-in Functions. Conditionals and Loops-if, else, elif, for, while, break, continue, pass**.**

Sequences: Strings, Lists, and Tuples- Built-in Functions, Special features.

Mapping and Set Types: Dictionaries, Sets.

Module –II **[9]**

Functions and Functional Programming –Calling Functions , Creating Functions, Passing Functions , Formal Arguments, Variable-Length Arguments, Functional Programming.

Modules–Modules and Files, Namespaces, Importing Modules, Module Built-in Functions, Packages, Related modules.

Module –III **[8]**

Files and Input / Output: File Objects, File Built-in Functions, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments.

Multithreaded Programming: Introduction, Threads and Processes, Python Threads, the Global Interpreter Lock, Thread Module, Threading Module.

Module -IV **[8]**

Graph terminology, Digraphs, Weighted graphs, Complete graphs, Graph complements, Bipartite graphs, Graph combinations, Isomorphism's, Matrix representations of graphs, incidence and adjacency matrices, Degree Sequence, Eulerian circuit: Konigsberg bridge problem, Touring a graph; Eulerian graphs,Hamiltonian cycles

Module – V **[7]**

Shortest paths: Dijkstra's algorithm, Walks using matrices.

Graph Coloring And Graph Algorithms: Four color theorem, Vertex Coloring, Edge Coloring, Coloring Variations.

Graph traversal: Depth-First Search, Breadth-First Search and its applications; The traveling salesman problem, Minimum Spanning Trees: Kruskal's and Prim's algorithm

TEXTBOOKS:

1. Core Python Programming, Wesley J. Chun, Third Edition, Prentice Hall PTR
2. Karin R Saoub, Graph Theory: An Introduction to Proofs, Algorithms, and Applications, 1 st edition, Chapman and Hall, 2021.

REFERENCES:

1. Think Python, Allen Downey, Green Tea Press
2. Introduction to Python, Kenneth A. Lambert, Cengage
3. Python Programming: A Modern Approach, VamsiKurama, Pearson
4. Learning Python, Mark Lutz, O'Really
5. R Balakrishnan, K Ranganathan, A Textbook of Graph Theory,Springer Exclusive, 2 nd edition, 2019.

24X0371: COMPUTER AIDED ENGINEERING GRAPHICS
(Common to All Branches)

B.Tech I Year II Semester

L	T	P	C
1	0	4	3

Course Overview:

Engineering Graphics is a foundational course designed to introduce first-year engineering students to the principles and practices of technical drawing and computer-aided design (CAD). This course covers essential topics such as geometric construction, orthographic projection, isometric drawing, lettering and dimensioning. Students will develop skills to create and interpret engineering drawings and gain proficiency in using CAD software for engineering applications.

Prerequisite: NIL

Course Objective: The students will be able

1. To understand the importance of engineering graphics in the engineering design process.
2. To apply principles of dimensioning and lettering in engineering drawings
3. To develop the ability to create and interpret technical drawings.
4. To master geometric constructions and projections.
5. To gain proficiency in computer-aided design (CAD) software.

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

1. Explain the role of engineering graphics in the engineering design and manufacturing process.
2. Understand the fundamental concepts of AutoCAD.
3. Perform basic geometric constructions and create accurate technical drawings.
4. Develop skills to create 2D and 3D drawings.
5. Use CAD software to create, modify, and manage engineering drawings.

Module-I: **Introduction to Engineering Graphics:** [12]

The Menu System, Toolbars (Standard, Object Properties, Draw, Modify and Dimension), Drawing Area (Background, Crosshairs, Coordinate System), Dialog boxes and windows, Shortcut menus (Button Bars), The Command Line, The Status Bar, Different methods of zoom as used in CAD, Select and erase objects.

Module-II: Conic Sections and Engineering Curves
[10]

Construction of Ellipse, Parabola, Hyperbola (General Method Only)
Engineering Curves: Cycloids, Epicycloid and Hypocycloid

Module -III: Orthographic Projections
[12]

Introduction to Projections: Assumptions, Principles and Different angles of projections.
Projections of Points: Located in all Quadrants
Projections of Lines: Parallel, Perpendicular, Inclined to one plane.

Module -IV: Projections of Planes and Projection of Solids [10]

Projections of Planes: Introduction to planes, Regular lamina- Orientations- Surface parallel to HP, Surface parallel to VP, Inclined to HP, Inclined to VP.

Projections of Solids: Introduction to solids, Right Regular Solids- Orientations- Axis perpendicular to HP, Axis perpendicular to VP, Axis inclined to HP, Axis inclined to VP.

Module –V: Isometric Drawing and Conversions

[14]

Principles of Isometric projections, Isometric View and Isometric Scale, Isometric view of: Planes and Solids, Conversion: Isometric to Orthographic and Vice Versa

Text Books:

1. **"Engineering Drawing",** N.D. Bhatt,Charotar Publishing House Pvt. Ltd, 53rd Edition, 2014, ISBN: 978-9380358173
2. **"Textbook of Engineering Drawing",** K. Venkata Reddy, BS Publications, Revised Edition, 2013, ISBN: 978-9381075994
3. **"Engineering Graphics**", K.R. Gopalakrishna, Subhas Stores, 32nd Edition, 2014, ISBN: 978-9353460206
4. "**Engineering Drawing and Computer Graphics",** M B Shah & C. Rana, Pearson Edition 2010.

Reference Books:

1. "**A Textbook of Engineering Drawing**", R.K. Dhawan, S. Chand Publishing, Revised Edition, 2012, ISBN: 978-8121914311
2. "**AutoCAD 2024: A Problem-Solving Approach, Basic and Intermediate**", Sham Tickoo, CADCIM Technologies, 1st Edition, 2023, ISBN: 978-1640571577
3. "**Engineering Drawing and Graphics Using AutoCAD**", T. Jeyapoovan, Vikas Publishing House 2nd Edition, 2015, ISBN: 978-9325982417

24X0072: ENGINEERING CHEMISTRY LAB

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 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 Thiokol rubber in the laboratory.
- Students will learn skills related to the lubricant properties such as saponification value, surface tension and viscosity of oils.

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

- Determination of parameters like hardness of water
- Able to perform methods such as conductometry, potentiometry and in order to find out the concentrations or equivalence points of acids and bases.
- Students are able to prepare polymers like bakelite and Thiokol rubber.
- Estimations saponification value, surface tension and viscosity of lubricant oils.

List of Experiments:

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

II. **Conductometry:** 1.Estimation of the concentration of an strong acid by Conductometry.

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

IV. **Dichrometry:** Determination of Ferrous ion by Dichrometry

V. **Preparations:**

1. Preparation of Thiokol rubber

VI. **Lubricants:**

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

VII. **Preparation of Hand sanitizer(Iso propyl alcohol)**

VIII. **Virtual lab experiments**

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

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).

(24X0271) Principles of Electrical and Electronics Engineering Lab
(Common to CSE, CSM and CSD)

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

COURSE OVERVIEW:
A element of electrical and electronics engineering laboratory offers a hands-on setting and simulation of basic circuits where students can apply the theoretical concepts and software packages learned in their electrical and electronics engineering courses. These labs are crucial for grasping the practical aspects of circuit design, electronics, electrical systems.

Pre-requisite: NIL

COURSE OBJECTIVES:
- To analyze a given network by applying various electrical and electronics laws and network theorems
- To know the response of electrical circuits for different excitations.
- To calculate, measure and know the relation between basic electrical parameters.
- To understand basic block sets of different simulation platform used in electrical/electronic circuit design.
- To understand use and coding in different software tools used in electrical/ electronic circuit design.
- To understand the simulation of electric machines/circuits for performance analysis.

COURSE OUTCOMES: Upon the completion of laboratory practical course, the student will be able to
- Get an exposure to basic electrical laws.
- Understand the response of different types of electrical circuits to different excitations.
- Understand the measurement, calculation and relation between the basic electrical parameters.
- Get an exposure to basic electronics devices and laws and Develop knowledge of software packages to model and program electrical and electronics systems.
- Model different electrical and electronic systems and analyze the results by used software packages for simulation in laboratory experimentation.

List of experiments / demonstrations:
PART A: Conduct All the experiments
1. Verification of Ohms Law.
2. Verification of KVL and KCL.
3. Verification of super position theorem.
4. Verification of Norton’s and Thevenin’s Theorem.
5. Resonance in Series RLC circuit.

PARTB: Simulate any five from following experiments using Multisim / MATLAB Software:

1. Simulate V-I Characteristics of PN Junction Diode in a)Forward Bias and b)Reverse Bias.
2. Simulate V-I Characteristics of Zener Diode and Observe Zener as a Voltage Regulator.
3. Simulate Characteristics of Half Wave Rectifier
4. Simulate Characteristics of Full Wave Rectifier.
5. Simulate the Performance Characteristics of a DC Shunt Motor.
6. Simulate the load test on single phase transformer to find out efficiency
7. Simulate Kirchhoff's voltage law using basic series DC Circuit - 4 with resistors. Where Vs = 6 V, R1 = 100 Ω, R2 = 220 Ω, R3 = 1k Ω
8. Simulate Kirchhoff's current law using basic parallel DC Circuits - 5 with resistors. Where Vs = 6 V, R1 = 100 Ω, R2 = 220 Ω, R3 = 1k Ω

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, 2 nd 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.

24X0572:Essentials Of Problem Solving Using Python Laboratory

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

Course Overview:

This course gives acquaintance to Python Programming and Graph Theory. It deals with Python programming concepts and concepts in graph theory like properties of standard graphs, Eulerian graphs, Hamiltonian graphs, Chordal graphs, Distances in graphs, Planar graphs, graph connectivity and Colouring of graphs. Python programming is used developing machine learning and data science applications. Graph theory is used in Network Topologies and Routing Algorithms, Algorithm Design, Social Network Design, Logistics.

Prerequisites:

- A course on “Problem solving using C and C++”.

Co-Requisites: Essentials of problem solving

Course Objectives: The students will try to learn

- Basic building blocks of python
- Using of Functions and Modules
- Importance of Multithreading in problem solving
- The fundamental concepts of graph theory
- Graph coloring and traversal algorithms for solving real-world problems

Course Outcomes: After successful completion of the course, students should be able to

- Construct Python data structures programs using tuples sets and dictionaries
- Design Programs using Functions and Modules
- Implement Multithread concept in solving problems
- Understand graph terminology
- Build efficient graph routing algorithms for various optimization problems ongraphs.

Week 1: Python Numbers

a) You are developing a program to determine whether a given year is a leap year, usingthe following formula: a leap year is one that is divisible by four, but not by one hundred, unless it is also divisible by four hundred. For example, 1992, 1996, and 2000 are leap years, but 1967 and 1900 are not. The next leap year falling on a century is 2400.

b) You are developing a program to determine the greatest common divisor and least common multiple of a pair of integers.

c) You are developing a program to create a calculator application. Write code that will take

two numbers and an operator in the format: N1 OP N2, where N1 and N2 are floating point or integer values, and OP is one of the following: +, -, *, /, %, **, representing addition, subtraction, multiplication, division, modulus/remainder, and exponentiation, respectively, and displays the result of carrying out that operation on the input operands.

Hint: You may use the string split() method, but you cannot use the exal () built-in function.

Skill Oriented Exercise

1. The cricket World Cup has started in Chefland. There are many teams participating in the group stage matches. Any team that scores 12 or more points in the group stage matches qualifies for the next stage.

2. The elections in Chefland have concluded, and the results are conducted. Chef received X votes, and his rival Chefu received Y. Chef thinks he dominated the election if and only if he received at least double the number of votes Chefu received. Did Chef dominate the election?

3. Bob has an account in the Bobby Bank. His current account balance is W rupees. Each month, the office in which Bob works deposits a fixed amount of X rupees to his account. Y rupees is deducted from Bob's account each month as bank charges. Find his final account balance after Z months. Note that the account balance can be negative as well.

4. You're a bit all over the place as a college student. You used to eat out at expensive restaurants almost every day until your parents gave you a talking-to about being irresponsible. Now, you've got to control your eating and spending habits. So, here's the plan: you'll stick to the college mess for your meals every day, except Sundays. On Sundays, you're treating yourself to those fancy restaurant meals. The cost is Rs.X for the mess food each day, and Rs. Y for the restaurant splurges. Now, what's the cost of food per week? Note that you don't have to pay for the mess on Sundays. (A week has seven days, as usual.)

Week 2: ***Control Flow***

a) Write a Program for checking whether the given number is a prime number ornot.
b) Write a program to print Fibonacci series upto given n value.
c) Write a program to calculate factorial of given integer number.
d) Write a program to calculate value of the following series $1+x-x^2+x^3-x^4+x^n$.
e) Write a program to print Pascal triangle.

Skill Oriented Exercise

1. Charlie is 17 years old and is eager to vote. Write a Python program to check if he meetsthe legal voting age of 18.

2. Your friend given a list of numbers to you and asked to find out the largest numberamong them. Write a python program to find the largest number.

3. Daemon don’t like the multiples of 7 so help him to write a Python program that printsnumbers from 1 to 30 but skip the number if it encounters multiple of a 7.

4. Bob has an account in the Bobby Bank. His current account balance is W rupees. Each month, the office in which Bob works deposits a fixed amount of X rupees to his account.Y rupees is deducted from Bob's account each month as bank charges. Find his final account balance after Z months. Note that the account balance can be negative as well.

Week-3 Python Sequences

a) Write a program to sort the numbers in ascending order and strings in reverse alphabetical order.
b) Given an integer value, return a string with the equivalent English text of each digit. For example, an input of 89 results in "eight-nine" being returned. Write a programto implement it.

c) Write a program to create a function that will return another string similar to the input string, but with its case inverted. For example, input of "Mr. Ed" will result in "mR.eD" as the output string.
d) Write a program to take a string and append a backward copy of that string, makinga palindrome.

Skill Oriented Exercise

1. Alice loves quotes. Write a Python program to count the number of characters in her favorite quote: "To be or not to be, that is the question.".
2. Emily wants to know if her friends name is a palindrome. Write a Python program to check for a name is a palindrome.
3. Charlie just read a new book and wants to add it to his set of favorite books {"The Hobbit", "Harry Potter"}. Write a Python program to add "The Great Gatsby" to Charlie's set and print the updated set.
4. You have a list of friends' ages: [25, 22, 29, 24]. Write a Python program to sort this listin ascending order.

Week-4 Python Dictionaries

a) Write a program to create a dictionary and display its keys alphabetically.
b) Write a program to take a dictionary as input and return one asoutput, but the valuesare now the keys and vice versa.
c) Given a List, extract all elements whose frequency is greater than K.Ex: Input test_list = [4,6,4,3,3,4,3,4,3,8], k=3

Output =[4,3]

Skill Oriented Exercise

1. You have a dictionary of your friends' favorite fruits: {"Alice": "Apple", "Bob": "Banana", "Charlie": "Cherry"}. Write a Python program to print Bob's favorite fruit.

2. John manages a small store and needs a program to track his product inventory. Write a Python program that will help John Creating the dictionary which contains the name and

price of the product and print the maximum product name along with its price.

Week-5 Files

a) Write a program to compare two text files. If they are different, give the line andcolumn numbers in the files where the first difference occurs.

b) Write a program to compute the number of characters, words and lines in a file.

Skill Oriented Exercise

1. Alice wants to list all files in her current directory. Write a Python program to import theos module and use it to print the names of all files in the current directory.
2. Charlie wants to read the contents of books.txt line by line and print each book name.Write a Python program to open the file and use the readline method to print each line.
3. Write a Python program greet.py that takes a name as a command-line argument andprints "Hello, [name]!". Demonstrate how to run it with the argument "Alice".

Week- 6&7 Functions

a) Write a function ball collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.
Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius
If (distance between two balls centers) <= (sum of their radii) then (they are colliding)
b) Find mean, median, mode for the given set of numbers in a list.
c) Write simple functions max2() and min2() that take two items and return the larger and smaller item, respectively. They should work on arbitrary Python objects. For example, max2 (4, 8) and min2(4, 8) would each return 8 and 4, respectively.

d) Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

e) Write a function dups to find all duplicates in the list.

f) Write a function unique to find all the unique elements of a list.

g) Write a function cumulative_ product to compute cumulative product of alist of numbers.
h) Write a function reverse to reverse a list. Without using the reverse function.
i) Write function to compute GCD, LCM of two numbers. Each functionshould n't exceed one line.

Skill Oriented Exercise

1. Alice wants to create a reusable function to greet her friends. Write a Python function greet that takes a friend's name as an argument and prints a greeting. Call the function with the name "Bob."

2. Charlie needs a function that can sum any number of arguments. Write a function sum_numbers that takes a variable number of arguments and returns their sum. Use thisfunction to sum 1, 2, 3, and 4.
3. Alice wants a quick way to increment a number by 1. Write a lambda function that takesa number and adds 1 to it. Use this lambda to increment 7.
4. Bob has written a custom module called mymath.py with a function add(a, b) that returns the sum of a and b. Write a Python program to import this module and use the add function to add 3 and 5.

Week- 8 Multithreading

a) Write a program to create thread using thread module.
b) Write a program to create thread using threading module.
c) Write a Program to use Python's threading module to calculate the square and cube of anumber concurrently.

Skill Oriented Exercise

1. Alice wants to perform two tasks simultaneously: counting numbers and printing messages. Write a Python program to create two threads, one for counting from 1 to 5 and another for printing "Hello" five times.
2. Charlie is learning about the Global Interpreter Lock (GIL). Write a Python program demonstrating how GIL affects multi-threaded CPU-bound tasks by incrementing a counter in two threads.
3. Diana wants to print numbers in a separate thread using the thread module. Write a Python program to print numbers from 1 to 5 in a new thread.
4. Emily needs a background thread to print a heartbeat message every second. Write a Python program to create a daemon thread that prints "Heartbeat" every second.
5. Alice is managing a shared resource. Write a Python program where two threads increment a shared counter using a threading.Lock to avoid race conditions.

Week 9:

a) Write a Python program to implement Euler Circuit.
b) Write a Python program to implement Dijkstra's algorithm.
c) Given a connected graph G with N nodes and M edges (edges are bi-directional). Every node is assigned a value $A[i]$. We define a value of a simple path as :

Value of path = Maximum of (absolute difference between values of adjacent nodes in a path). A path consists of a sequence of nodes starting with start node S and end node E.

$S-u1-u2-....-E$ is a simple path if all nodes on the path are distinct and $S,u1,u2,...,E$ are nodes in G.

Given a start node S and end node E, find the minimum possible ***"valueofpath"*** which starts with node S and ends with node E.

d) Yatin created an interesting problem for his college juniors. Can you solve it?

Given N rooms, where each room has a one-way door to a room denoted by $room[i]$, where $1 \leq i \leq N$. Find a positive integer K such that, if a person starts from room i, $(1 <= i <= N)$, and continuously moves to the room it is connected to (i.e. $room[i]$), the person should end up in room i after K steps;

Note: The condition should hold for each room.If there are multiple possible values of K modulo (109+7), find the smallest one.If there is no valid value of K, output −1

Week 10: Implement the following using python

a) M-coloring
b) Vertex coloring
c) Edge coloring

Week 11: Implement the following graph traversal methods.

a) Depth-First Search
b) Breadth-First Search
c) You are presented with a network comprising N computers and M wired connections between them. Your objective is to optimize the network's connectivity using precisely K wires from your inventory. The aim is to **maximize** the number of computers that can be linked together within the given constraints. Your task is to determine and report the size of the largest network that can be formed by establishing these connections.

In the context of this problem, computers are considered connected if they share either a direct or indirect wired connection. It is worth noting that the value of K will always be less than the number of isolated (standalone) networks in the given configuration, and it may even be zero.

d) A country consists of N cities. These cities are connected with each other using $N-1$ bidirectional roads that are in the form of a tree. Each city is numbered from 1 to N. You want to safeguard all the roads in the country from any danger, and therefore, you decide to place cameras in certain cities. A camera in a city cansafeguard all the roads directly connected to it. Your task is to determine the minimum number of cameras that are required to safeguard the entire country.

Week 12: Travelling Salesman problem**.**

a) You are working in a salesmen company as a programmer.

There are *n* towns in your country and *m* directed roads between them. Each road has a cost person should spend on fuel. The company wants to sell goods in all *n* towns. There are infinitely many salesmen in the company. We can choose some positive number of salesmen and give a non-empty list of towns to each of them. Towns from the list are the towns to sell goods in. Each salesman will visit all the towns in his list in this particular order in cycle (after the last town he will return to the first town and so on). Salesman can visit other towns on his way but he will not sell goods in these towns. Two salesmen cannot sell goods in one town because it will attract unnecessary attention to your company. But for every town there must be a salesman who sell goods in this town. If salesman's list of towns consists of exactly one town then he should pay fee to stay in this town each month (each town has its own fee) or he should go for a round trip and spend money on fuel.

Your task is to calculate the minimal amount of money company must spend monthly to achieve its goals. We will assume that every salesman will spend a month to make one cycle.

b) It is the final leg of the most famous amazing race. The top 'n' competitors have made itto the final. The final race has just begun. The race has 'm' checkpoints. Each team can reach any of the 'm' checkpoint but after a team reaches a particular checkpointthat checkpoint gets closed and is not open to any other team. The race ends when 'k' teams finish the race. Each team travel at a constant speed throughout the race which might be different for different teams. Given the coordinates of n teams and m checkpoints and speed of individual team return the value of minimum time needed to end the race.

c) Little Jhool is a very lenient teaching assistant in his college. He doesn't like cutting the marks of students, so obviously, every student in his tutorial loves him. But anyway, the teacher has got to know about the leniency of Jhool while giving marks, so this time in exam, he decides to give a different exam paper to every single student to check how well have the students been taught by Jhool. Now, Little Jhool knows the strong and weak topics of every single student, so he wants to maximize the total marks obtained by students in his tutorial.You are given the number of students in Jhool's tutorial, denoted by **n** - **n** also being the number of different exam papers - that is, one for every student. Every student will get only one exam paper to solve. You are further given a matrix, *(n x n)* denoting the marks every student will get if he attempts a particular exam paper. You've to help Jhool figure out a way by which he could maximize the total score obtained by his entire class.

Week 13: Construct minimal spanning tree using the following

a) Prim's Algorithm
b) Kruskal's Algorithm
c) There are total N Hacker-cities in a plane. Each city is located on coordinates ($X[i]$,$Y[i]$) and there can be any number of cities on the same coordinates.

You have to make these cities connected by constructing some roads in such a waythat it is possible to travel between every pair of cities by traversing the roads. The

cost of constructing one road between any two cities is the minimum of the absolute difference between their X and Y coordinates.

As you want to earn more and more, you decided to do this in the most optimal way possible, such that the total cost of constructing these roads is minimal. You have to return the minimum money you need to spend on connecting all the cities.

d) Tom is visiting the country Hackerland. Hackerland has *n* cities and *m* bi-directional roads. There are *k* types of tokens. Token *i* costs ci. The costs of the tokens are such that for all $2 \le i \le k$, $ci \ge 2ci-1$. For each road, you need to have a particular set of tokens, if you want to travel it. Note that you don't have to give the tokens, you just need to show them. Thus, one token can be used at any number of roads, where it is required. Tom wants to select a set of tokens, such that using them, he can go from any city to any other city. You have to help him minimize the total cost of tokens he buys.

TEXT BOOKS:

1. Core Python Programming, Wesley J. Chun, Third Edition, Pearson.
2. Karin R Saoub, Graph Theory: An Introduction to Proofs, Algorithms, and Applications, 1 st edition, Chapman and Hall, 2021.

REFERENCE BOOKS:

1. Think Python, Allen Downey, Green Tea Press
2. Introduction to Python, Kenneth A. Lambert, Cengage
3. Python Programming: A Modern Approach, VamsiKurama, Pearson
4. Learning Python, Mark Lutz, O'Really.

24X0027: PUBLICSPEAKING SKILLS

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

Course Description: This course is designed to develop students' public speaking skills, focusing on speech preparation, delivery techniques, and the use of non-verbal communication. Students will learn to present effectively in various contexts, from formal presentations to informal meetings.

Prerequisites: Basic communication skills

Course Objectives: The students will learn:

- Understand the fundamentals and prerequisites of public speaking.
- Develop the ability to convert ideas into structured speeches.
- Enhance performance through verbal and non-verbal communication.
- Master different types of public speaking and professional presentations.
- Learn the etiquette and manner isms required for effective public speaking.

Course Outcomes: By the end of this course, students will be able to:

1. **Prepareanddeliver**speechesconfidentlybyorganizingcontentandovercomingstagefright.
2. **Utilize** verbal and non-verbal communication to engage the audience effectively.
3. **Adapt** to different contexts by tailoring speeches for diverse audiences and settings.
4. **Exhibit**professionalismandcreativityusingproperetiquette,rhetoricaldevices,andcreative language

Module1: Fundament also f Public Speaking(Noofhours:6)

This unit introduces public speaking, covering its importance, course structure, assessment methods, and prerequisites like understanding the audience, research, organizing speech structure, over coming Stage fright, and practicing speech delivery.

Module2:SpeechDevelopmentandDelivery(Noofhours:6)

This unit focuses on converting ideas into action through brainstorming, outlining, drafting speeches, using storytelling, and practicing impromptu speaking. It also emphasizes public speaking as a performative act, covering voice modulation, eye contact, audience engagement, effective pacing, gestures, and techniques for handling questions and interruptions.

Module3: Non-verbal Communication and Speech Types(Noofhours:5)

This unit explores non-verbal communication's role in public speaking, covering the importance and types of cues like facial expressions and gestures, aligning verbal with non-verbal messages, observing audience feedback, and adapting non-verbal communication to virtual settings. It also addresses various types of public speaking, including informative, persuasive, special occasion, motivational speeches, panel discussions, and debates.

Module4: Professional and Formal Speaking(Noofhours:6)

This unit covers speeches, including analysis of famous examples, preparing and delivering various types, conducting peer and self-evaluation, and effectively utilizing visual aids and multimedia. Adapting speeches for diverse audiences, and addressing practical skills for interviews, professional communication,conductingmeetings,conferences,presentations,andbuildingprofessionalnetworks.

Module5:Advanced Techniques and Professionalism(Noofhours:5)

This unit focuses on structuring and delivering professional presentations effectively, using creative language techniques for impactful messaging, and embodying proper etiquette and professionalism in public speaking.

TEXTBOOK:

- **"The Art of Public Speaking**" by Dale Carnegie Prabhat Prakashan Pvt. Ltd.; First Edition (31 December 2020) **ISBN-10:**8184302614

REFERENCEBOOKS:

- **"The Art of Public Speaking" by Stephen E. Lucas**, ISBN: 978-0073523910, Year of Publication: 2014, Publisher: McGraw-Hill Education
- **Confessions of a Public Speaker" by Scott Berkun**, ISBN: 978-0596801991, Year of Publication: 2010, Publisher: O'Reilly Media ISBN: 978-0596801991, Year of Publication: 2010
- **"Speak Like Churchill, Stand Like Lincoln: 21 Powerful Secrets of History's Greatest Speakers" by James C. Humes,** ISBN: 978-0761563518, Year of Publication: 2002, Publisher: Three Rivers Press
- **"The Quick and Easy Way to Effective Speaking" by Dale Carnegie,** ISBN: 978-0671724009, Year of Publication: 1990, Publisher: Pocket Books

(R24 – CSM) II Year Course Structure and Syllabus Applicable From 2024-25 Admitted Batch

II YEAR I SEMESTER (III SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2436703	Discrete Mathematics	PC	2	0	0	2	40	60	100
2	2430510	Operating systems	PC	3	0	0	3	40	60	100
3	2430505	Software Engineering	PC	2	0	0	2	40	60	100
4	2430005	Mathematical and Statistical Foundations	BS	3	1	0	4	40	60	100
5	2430507	Data Structures	ES	3	0	0	3	40	60	100
	243ExL1	Design and Innovation	EL	0	0	2	1	40	60	100
		Laboratory								
1	2430574	Data visualization-PowerBI	PC	0	0	2	1	40	60	100
2	2430577	Operating systems Laboratory	PC	0	0	2	1	40	60	100
3	2430575	Data Structures laboratory Using Python	ES	0	0	2	1	40	60	100
	2430588	Internship-1 *	PS	0	0	2	1	100	-	100
		Skill Development								
1	2430598	Parallel Computation - RUST	SDC	0	0	2	1	40	60	100
		Mandatory Course								
1		Indian Knowledge System*	MC	0	0	0	0	-	-	-
Total Credits				**13**	**1**	**12**	**20**	**500**	**600**	**1100**

- **Students can choose any one of the following course**
  - i) 24XIKS1:Indian Science, Engineering and Technology
  - ii) 24XIKS2:Fundamentals and Applications of Vedic Mathematics
  - iii) 24XIKS3:Indian Health, Wellness and Psychology- including Ayurved
  - iv) 24XIKS4:Indian Town Planning and Architecture

****students have to complete internship in I year-II semester summer break with minimum 2 Weeks duration.**

II YEAR II SEMESTER (IV SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2440512	Artificial Intelligence	PC	3	0	0	3	40	60	100
2	2440503	Database Management Systems	PC	3	0	0	3	40	60	100
3	2446606	Computer System Architecture	PC	3	1	0	4	40	60	100
4	2440508	Design and Analysis of Algorithms	PC	2	0	0	2	40	60	100
5	2440511	OOPS through JAVA	ES	3	0	0	3	40	60	100
6	244EXL2	Prototype/model development and Entrepreneurship	EL	0	0	2	1	40	60	100
		Laboratory								
1	2440581	Artificial Intelligence Laboratory	PC	0	0	2	1	40	60	100
2	2440573	Database Management Systems Laboratory	PC	0	0	2	1	40	60	100
3	2440576	Design and Analysis of Algorithms through JAVA Laboratory	ES	0	0	2	1	40	60	100
		Skill Development								
1	2440597	NO SQL Data Bases (MONGO DB)	PS	0	0	2	1	40	60	100
		Mandatory Course								
1	2440021	Environmental Science	MC	0	0	0	0	-	-	-
Total Credits				14	1	10	20	400	600	1000

II – I

2436703: DISCRETE MATHEMATICS

B.Tech. II Year I – Sem.

L	T	p	C
2	0	0	2

Course Objectives: The students will try to learn

- Concepts of mathematical logic.
- The concepts of sets, relations, and functions.
- Algebraic structures
- Perform the operations associated with sets, functions, and relations.
- Recurrence relations.

Course Outcomes: After successful completion of the course, students should be able to

- Construct precise mathematical proofs
- Understand sets, relations, functions, and discrete structures.
- Analyze and solve counting problems on finite and discrete structures
- Use logical notation
- Formulate problems and solve recurrence relations.

MODULE – I **8**

Mathematical logic: Introduction, Statements and Notation, Connectives, well formed formula, Equivalence of formulas, Normal forms, Theory of inference for the statement calculus, predicate calculus, Inference theory of predicate calculus.

MODULE – II **7**

Set theory: Basic concepts of set theory, Set and Operations on sets, Relations and ordering, properties of binary relations in a set, Equivalence relation, Compatibility of relation, partial order relation, partial order set, Functions, Composition of functions, Inverse function, Recursive functions.

Module - III **6**

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

MODULE-IV **7**

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and permutations, Binomial Coefficients, Binomial and Multinational Theorems, Principle of Inclusion-Exclusion.

MODULE-V **7**

Recurrence Relations: Generating Functions of Sequences, Calculating Coefficients of generating functions, Recurrence relations, Solving recurrence relations by substitution and generating functions, Method of Characteristic roots, Solutions of Inhomogeneous Recurrence Relations.

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.

2430510: OPERATING SYSTEMS

B.Tech. II Year I – Sem.

L	T	P	C
3	0	0	3

Course Overview:

Operating Systems course provides theoretical knowledge about the structure of operating systems, process, memory management and virtual memory implementation principles, input-output management and deadlock avoidance, file system structure. The Operating System manages the computer's software and hardware as well as its memory and processes. Computer operating systems also allow users to see information, create and save files, and use applications such as email and web browsers. Various applications of operating systems include security, job accounting, error detection aids, coordination between other software's and users.

Prerequisites:

- A course on Problem solving Using c and c++

Course Objectives: The students will try to learn

- Describes functionalities of main components in operating systems.
- Analyze the algorithms used in process management.
- Gives synchronization and deadlock concepts.
- Analyze the algorithms used in memory management.
- Interpret the concepts of input and output storage for file management

Course Outcomes: After successful completion of the course, students should be able to

- Control accesses a computer and the files that may be shared
- Gain knowledge on process and Process Scheduling
- Understand Deadlock Prevention, Deadlock Handling and Synchronization
- Apply Memory Management techniques in OS.
- Understand File System.

Module-I **[10]**

Operating System Introduction: What is an operating system do, computer system organization, computer system architecture, operating system structure- operating system operations, process management, memory management, operating system services, System Calls, types of system calls.

Module –II **[9]**

Process: process concepts, process scheduling, operations on processes, Interprocesses communication, multithreading models, thread libraries.

Process Scheduling: Scheduling criteria, scheduling algorithms, thread scheduling Multiple- Processor Scheduling.

Module –III **[9]**

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

Synchronization: background, the critical section problem, peter's solution, Synchronization hardware, semaphores, Classical Problems of Synchronization, Monitors.

Module –IV **[8]**

Memory Management and Virtual Memory –background, Swapping, Contiguous Allocation, Paging structure of the page table, Segmentation.

Virtual memory: background, demand paging page replacement allocation of frames thrashing.

Module –V **[8]**

File System : File system and implementing file system, file concept access methods, directory and file system structure, File system implementation, Directory implementation, Allocation methods, Free-space Management, efficiency and performance, recovery, NFS.

TEXT BOOKS:

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

REFERENCES:

1. Operating Systems – Internals and Design Principles, Stallings, 5th Edition, Pearson Education/PHI,2005.
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

2430505: SOFTWARE ENGINEERING

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

Course Description: This course introduces the characteristics of software Generic view of software and the process models. The course emphasizes on software requirements system models, UML Basics, testing strategies, Metrics for process & products, instruction set design, RISK Management, RISK Identification, Quality Management, ISO : 9000 Quality

Pre-requisite: NIL

Course Objectives: **The students will try to learn**

- Importance of software engineering principles and software process framework
- Contemporary approaches for design models and requirements validation
- Various metrics and quality assurance strategies
- Designing of testing report.
- Different strategies for testing and risk management

Course Outcomes: After successful completion of the course, students should be able to

- Understand different process models
- Translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
- Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
- Develop a simple testing report
- Understand basics of Risk Management and Quality Management

Module –I **9**

INTRODUCTION TO SOFTWARE ENGINEERING: The Evolving Role of Software, Characteristics of Software, The Changing Nature of Software, Legacy Software, Software Myths.

A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models.

Process models: The waterfall model, incremental process models, evolutionary process models, the unified process, Agile models: Extreme Programming, Scrum, DSDM, FDD, CRYSTAL and Lean Software Development.

Module –II **8**

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.

Module –III **8**

System models: Context models, behavioral models, data models, object models, structured methods Design Engineering: Design process and design quality, design concepts, the design model, software architecture, Architectural styles and patterns. Introduction to UML: Basic Building Blocks of UML- Things, Relationships and Diagrams.

Module –IV **9**

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

Metrics for Process and Products: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance, metrics for software quality.

Module –V **8**

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, software reliability, the ISO 9000 quality standards.

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.
   The unified modelling language user guide Grady Booch, James Rambaugh, Ivar
3. Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pe

2430005: MATHEMATICAL AND STATISTICAL FOUNDATIONS

(CSM, CSD)

II Year B.Tech. I Sem

L	T	P	C
3	1	0	4

Course Overview:

The student achieves the knowledge to analyze the problems using the probability and statistics. This course develop a skills in analyzing statistical data, skills in mathematical expectation, probability distribution, test of hypothesis for problems in engineering. Statistical models are used to identify, analyze and quantify potential risk through probability theory. It enables engineers to understand the risk associate with particular activities or situations, allowing them to take more effective steps to migrate them.

Pre-requisites: Mathematics courses of first year of study

Course Objectives: The student will try to learn

- The Concept of Random variables.
- Expectation and correlation.
- Probability distributions of single random variables.
- The sampling theory and the concept of Estimation.
- Testing of hypothesis and making statistical inferences.

Course outcomes: After successful completion of the course, students should be able to

CO1: Formulate and solve real world problems involving Random variables.
CO2: Correlate the concepts of one unit to the concepts in other units.
CO3: Identify probability distributions to various case studies.
CO4: Understand the concept of sampling and apply concept of Estimation.
CO5: Apply the concept of testing a hypothesis to case studies.

UNIT-I: Probability and Random Variables **10 L**

Sample space, Events, Counting sample points, Probability of an Event, Additive Rules, Conditional Probability, Independent events, Product Rule and Baye's Rule (All rule without proof). Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions.

UNIT-II: Mathematical Expectation and Correlation **10L**

Expectation and Mean of a Random Variable, Variance and Covariance of Random Variables, Means and variances of Linear combinations of random variables, Marginal, Conditional and Joint Probability distributions. Statistical Independence. Correlation, Karl Pearson Correlation Coefficient and Spearman's rank correlation Coefficient.

UNIT-III: Probability distributions **10L**

Discrete Probability distributions: Binomial and Poisson distributions, statistical parameters for these distributions (without proof).

Continuous probability distributions: Normal and Uniform distributions, statistical parameters for these distributions (without proof).

UNIT-IV: Sampling Distribution & Estimation **8L**

Sampling Distributions: Random Sampling, Parameters and Statistics, Sampling Distribution of Means and the Central Limit Theorem, t - Distribution, F-Distribution and Chi-square distribution.

Estimation: Estimating the Mean, Standard Error of a Point Estimate, Confidence Intervals for single sample and two Samples.

UNIT-V: Test of Hypothesis **10L**

10L

Statistical Hypothesis, General Concepts, Testing a Statistical Hypothesis, Tests Concerning Large samples - Single Mean, Two Means, Single Proportion and two Proportions. Small samples- T-test for Single Mean and Two Means, Chi-square test for independence of attributes.

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.

2430507: DATA STRUCTURES
(COMMON TO ALL BRANCHES)

B.Tech. II Year I Sem.

L	T	P	C
3	0	0	3

Course Overview:

Data structures are the fundamental building blocks of computer programming. They define how data is organized, stored, and manipulated within a program. Understanding data structures is very important for developing efficient and effective algorithms. In this Course, student will explore the most commonly used data structures, including **linked lists, stacks, queues, trees, and Hashing.**

Prerequisites:

- A course on "Essentials of Problem Solving using python".

Course Objectives: The students will try to learn

- Various linear and non-linear data structures.
- How to perform operations on data structures.
- Priority Queues and Heaps
- Various searching and sorting techniques.
- Different hashing techniques

Course Outcomes: After successful completion of the course, students should be able to

- Develop solutions by using different types of linked lists
- Solve problems using stack and queue
- Learn different types of trees and their applications
- Implement and know the application of algorithms for searching and sorting.
- Design Programs using Hashing

Module -I: Linked Lists **[9]**

Introduction to Data Structures, Linear list – singly linked list, Doubly linked list, Circular linked list - operations and its applications.

Module-II: Stack and Queue **[8]**

Stacks- Introduction, Operations, array and linked representations of stacks, stack applications (Infix to postfix conversion and postfix evaluation),

Queues- Introduction, operations, array and linked representations of queues and its applications.

Module-III: Trees **[10]**

Trees: General Trees, Binary Trees, Implementing Trees, Tree traversals

Search Trees: Binary Search Trees,Balanced search trees- AVL trees, B- trees

Priority Queue and Heaps: Priority queue ADT, Priority queue, Applications, Heap Trees, implementing a priority queue with a Heap, Heap Sort.

Module–IV: Searching and Sorting [9]

Searching: Linear Search and Binary Search and its applications.

Sorting: Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort and its applications.

Module – V: Hashing [8]

Introduction, Hash Functions-Modulo, Middle of Square, Folding, Collision Resolution Techniques-Separate Chaining, Open addressing,- Linear Probing, Quadratic Probing, Double Hashing.

TEXTBOOKS:

1. Fundamentals of data structures in C, E.Horowitz, S.Sahni and Susan Anderson Freed, 2nd Edition, Universities Press.
2. Data structures using C, A.S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/pearson education.

REFERENCES:

1. Data structures: A Pseudocode Approach with C, R.F.Gilberg And B.A.Forouzan, 2nd Edition, Cengage Learning.
2. Introduction to data structures in C, Ashok Kamthane, 1st Edition, PEARSON.

24X0574: DATA VISUALIZATION - POWER BI

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

Course Overview:

This course deals with report design and formatting in Power BI, which offers extraordinary visuals for building reports and dashboards. Additionally, gives acquaintance how to use report navigation to tell a compelling, data-driven story in Power BI.

Prerequisites: Nil

Course Objectives: The students will try to learn

- Importing of data from various sources.
- PowerBI Concepts
- Mapping of Visual Layouts and Graphical Properties.
- How to create Dashboard using PowerBI
- Developing of charts using PowerBI.

Course Outcomes: After successful completion of the course, students should be able to

- Understand How to import data into Power BI
- Understand Power BI concepts of Dimensions and Measures.
- Develop Programs and understand how to map Visual Layouts and Graphical Properties.
- Create a Dashboard that links multiple visualizations.
- Use graphical user interfaces to create Frames for providing solutions to real world problems.

List of Experiments:

1. Understanding Data, What is data, where to find data, Foundations for building Data Visualizations, Creating Your First visualization?
2. Getting started with Power BI Software using Data file formats, connecting your Data to Power BI , creating basic charts(line, bar charts, Tree maps),Using the Show me panel.
3. Power BI 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 Power BI data, Pivoting Power BI data.
6. Structuring your data, Sorting and filtering Power BI data, Pivoting Power BI 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, adding interactivity to your Dashboard, Distributing & Publishing your Visualization.
9. Power BI file types, publishing to Power BI Online, Sharing your visualizations, printing, and Exporting.
10. Creating custom charts, cyclical data and circular area charts, Dual Axis charts.

REFERENCES:

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.

2430577: OPERATING SYSTEMS LAB

B.Tech. II Year I Sem.

L	T	P	C
0	0	2	1

Prerequisites:

- A course on "Programming for Problem Solving"
- A course on "Data Structures".

Course Outcomes: The students will be able to:

- Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
- Implement C programs using Unix system calls.

LISTOFEXPERIMENTS

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

TEXTBOOKS:

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.

REFERENCEBOOKS:

1. Operating Systems – Internals and Design Principles 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

2430575: DATA STRUCTURES LAB USING PYTHON

B.Tech. II Year I Sem.

L	T	P	C
0	0	2	1

Course Overview:

Data structures are the fundamental building blocks of computer programming. They define how data is organized, stored, and manipulated within a program. Understanding data structures is very important for developing efficient and effective algorithms. In this Course, student will explore the most commonly used data structures, including linked **lists, stacks, queues, trees, and Hashing.**

Prerequisites:

- A course on “Essentials of Problem Solving”.

Course Objectives: The students will try to learn

- Various linear and non-linear data structures.
- How to perform operations on data structures.
- Priority Queues and Heaps
- Various searching and sorting techniques.
- Different hashing techniques

Course Outcomes: After successful completion of the course, students should be able to

- Develop solutions by using different types of linked lists
- Solve problems using stack and queue
- Learn different types of trees and their applications
- Implement and know the application of algorithms for searching and sorting.
- Design Programs using Hashing

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
2. Write a program that uses functions to perform the following operations on doubly linked list.: i) Creation ii) Insertion iii) Deletion
3. Write a program that uses functions to perform the following operations on circular linked list: i) Creation ii) Insertion iii) Deletion
4. Write a program that implement stack operations using i) Arrays ii) Pointers
5. Write a c program to implement infix to postfix conversion using stack.
6. Write a c program to implement postfix evaluation.
7. Write a program that implement Queue operations using i) Arrays ii) Pointers
8. Write a program to implement the tree traversal methods using both recursive and non-recursive.

9. Write a program to implement tree operations on i) AVL Trees ii) B Trees iii) Heap
10. Write a program that implements the following sorting methods to sort a given list of integers in ascending order i) Bubble sort ii) Selection sort iii) Insertion sort
11. Write a program that implements the following sorting methods to sort a given list of integers in ascending order i) Merge sort ii) Quick sort iii) Heap Sort
12. Write a program that use both recursive and non-recursive functions to perform the following searching operations for a Key value in a given list of integers: i) Linear search ii) Binary search
13. Write a program to implement hashing.

CASE STUDY-1 Balanced Brackets

A bracket is considered to be any one of the following characters: (,), {, }, [, or].

Two brackets are considered to be a *matched pair* if the an opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) *of the exact same type*. There are three types of matched pairs of brackets: [], {}, and ().

A matching pair of brackets is *not balanced* if the set of brackets it encloses are not matched. For example, {[(])} is not balanced because the contents in between { and } are not balanced. The pair of square brackets encloses a single, unbalanced opening bracket, (, and the pair of parentheses encloses a single, unbalanced closing square bracket,].

By this logic, we say a sequence of brackets is *balanced* if the following conditions are met:

- It contains no unmatched brackets.
- The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets.

Given strings of brackets, determine whether each sequence of brackets is balanced. If a string is balanced, return YES. Otherwise, return NO.

CASE STUDY-2 Minimum Average Waiting Time

Mr. Raju owns a pizza restaurant and he manages it in his own way. While in a normal restaurant, a customer is served by following the first-come, first-served rule, Raju simply minimizes the average waiting time of his customers. So he gets to decide who is served first, regardless of how sooner or later a person comes.

Different kinds of pizzas take different amounts of time to cook. Also, once he starts cooking a pizza, he cannot cook another pizza until the first pizza is completely cooked. Let's say we have three customers who come at time t=0, t=1, & t=2 respectively, and the time needed to cook their pizzas is 3, 9, & 6 respectively. If Raju applies first-come, first-served rule, then the waiting time of three customers is 3, 11, & 16 respectively. The average waiting time in this case is (3 + 11 + 16) / 3 = 10. This is not an optimized solution. After serving the first customer at time t=3, Raju can choose to serve the third customer. In that case, the waiting time will be 3, 7, & 17 respectively. Hence the average waiting time is (3 + 7 + 17) / 3 = 9.

Help Raju achieve the minimum average waiting time. For the sake of simplicity, just find the integer part of the minimum average waiting time.

Note:

- The waiting time is calculated as the difference between the time a customer orders pizza (the time at which they enter the shop) and the time she is served.
- Cook does not know about the future orders.

TEXTBOOKS:

1. Fundamentals of data structures in C, E.Horowitz, S.Sahni and Susan Anderson Freed, 2nd Edition, Universities Press.
2. Data structures using C, A.S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/pearson education.

REFERENCES:

1. Data structures: A Pseudocode Approach with C, R.F.Gilberg And B.A.Forouzan, 2nd Edition, Cengage Learning.
2. Introduction to data structures in C, Ashok Kamthane, 1st Edition, PEARSON

2430598: PARALLEL COMPUTATION: RUST

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

Module I – Basics of Rust

1. **Hello, World!**
   - Basic Rust program structure using `fn main()`.
2. **Variable Declaration and Mutability**
   - Using `let`, `mut`, and type inference.
3. **Data Types Demo**
   - Use `i32`, `f64`, `char`, `bool`, etc., in a single program.
4. **Arithmetic and Logical Operators**
   - Calculator-style program showing arithmetic and logical operations.
5. **Using `cargo` for a Simple Project**
   - Create a simple project using `cargo new`, build, and run it.

Module II – Ownership and Control Flow

6. **Ownership Transfer**
   - Pass variables to functions and explore ownership and `move`.
7. **Borrowing and References**
   - Demonstrate `&T` and `&mut T` references and restrictions.
8. **Lifetimes Example**
   - Use lifetime annotations in a function returning references.
9. **If-Else Statement**
   - Even/odd checker or grade calculator.
10. **Looping Examples**
    - Use `for`, `while`, and `loop` to iterate over arrays or counters.
11. **Pattern Matching with `match`**
    - Match numbers to print weekdays or match enums.
12. **Nested Control Statements**
    - Combine loops and conditionals in one program (e.g., a simple number guessing game).

Module III – Functions and Structs

13. **Functions with Arguments and Return Values**
    - Create a function to calculate factorial or square of a number.

14. **Ownership in Functions**
    - Pass by value vs. reference in function parameters.
15. **Using Structs**
    - Define a `Rectangle` struct and calculate area.
16. **Tuple Structs and Field Init Shorthand**
    - Define a `Color(u8, u8, u8)` and use field shorthand for init.
17. **Enums and Pattern Matching**
    - Define an `enum` for `TrafficLight` and match its values.
18. **Option Enum Usage**
    - Safe division function returning `Option<f64>`.

Module IV – Project Management and Smart Pointers

19. **Error Handling with Result**
    - File reading or division with error handling using `Result`.
20. **Using Box, Rc, and RefCell**
    - Demonstrate smart pointers with a simple linked list or counter program.

Module- V – Object oriented Programming

21. **"Polymorphic Behavior with Trait Objects: A Speakable Animal Zoo"**
Covers: Traits, dynamic dispatch, trait objects (`&dyn Trait`)

22. **"Implementing Strategy Pattern with Traits for Payment Processing"**
Covers: Object-oriented design using traits and `Box<dyn Trait>` for dynamic strategy switching

23. **"Trait-Based Drawing Application with Heterogeneous UI Components"**
Covers: Object-oriented characteristics, trait objects, and allowing different drawable types in a single collection

II-II

2440512: ARTIFICIAL INTELLIGENCE

B.Tech. II Year II Sem.

L	T	P	C
3	0	0	3

Course Overview:

This course aims to introduce the fundamental concepts of artificial intelligence (AI). Students will develop a broad understanding of AI technologies, their implications, and their potential applications in various fields. The course will emphasize practical examples and real-world case studies to facilitate comprehension and inspire innovative thinking.

Prerequisites:

- Data Structures, Discrete Mathematics

Course Objectives: The students will try to learn

- The fundamental concepts and subfields of AI.
- Real-world applications of AI across various industries.
- The knowledge base and application of reasoning
- First-order logic to solve real world problems..
- Recognize the potential of AI to drive innovation and transformation in different domains.

Course Outcomes: After successful completion of the course, students should be able to

- Frame an efficient problem space for a problem expressed in natural language.
- Finalize a search algorithm for a problem and estimate its time and space complexities.
- Know the knowledge base and application of reasoning
- Apply first-order logic to solve real world problems.
- Possess the skill for representing knowledge using the appropriate technique for a given problem.

Module-I **[9]**

Artificial Intelligence: What is AI, Foundations and History of AI. Propositional and first order logic.

Intelligent Agents: Introduction, how Agents Should Act, Structure of Intelligent Agents, Agent programs, Simple reflex agents, Goal based agents, Utility based agents, Environments and Environment programs.

Problem Solving by Search: Problem-Solving Agents, Formulating Problems, Example Problems, Searching for Solutions, Search Strategies (Breadth-first search, Uniform cost search, Depth-First Search, Iterative deepening Depth-First search, Bidirectional search).

Module –II **[8]**

Informed Search Methods: Best-First Search, Heuristic Functions, Memory Bounded Search, Iterative Improvement Algorithms**.**

Game Playing: Introduction, Games as Search Problems, Perfect Decisions in Two-Person Games, Imperfect Decisions, Alpha-Beta Pruning, Games That Include an Element of Chance, State-of-the- Art Game Programs.

Module –III **[9]**

Knowledge and Reasoning: A Knowledge-Based Agent, The Wumpus World Environment, Representation, Reasoning, and Logic, Prepositional Logic, An Agent for the Wumpus World.

First-Order Logic: Syntax and Semantics, Extensions and Notational Variations, Using First-Order Logic, Logical Agents for the Wumpus World, A Simple Reflex Agent, Representing Change in the World Building a Knowledge Base: Properties of Good and Bad Knowledge Bases, Knowledge Engineering, The Electronic Circuits Domain, General Ontology, Application: The Grocery Shopping World.

Module –IV **[10]**

Inference in First-Order Logic: Inference Rules Involving Quantifiers, An Example Proof, Generalized Modus Ponens, Forward and Backward Chaining, Resolution: A Complete Inference Procedure, Completeness of resolution.

Logical Reasoning Systems: Introduction, Indexing, Retrieval, and Unification, Logic Programming Systems, Theorem Provers, Forward-Chaining Production Systems, Frame Systems and Semantic Networks, Description Logics, Managing Retractions, Assumptions, and Explanations

Module –V **[9]**

Planning: A Simple Planning Agent, From Problem Solving to Planning, Planning in Situation Calculus, Basic Representations for Planning, A Partial-Order Planning Example, A Partial-Order Planning Algorithm, Knowledge Engineering for Planning.

Practical Planning: Practical Planners, Hierarchical Decomposition, Analysis of Hierarchical Decomposition, Resource Constraints.

TEXTBOOKS:

1. Artificial Intelligence A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson Education

REFERENCES:

1. Artificial Intelligence, E.Rich and K.Knight, , 3rd Edition, TMH
2. Artificial Intelligence, Patrick Henny Winston, 3rd Edition, Pearson Education.
3. Artificial Intelligence, ShivaniGoel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

2440503: DATABASE MANAGEMENT SYSTEMS

B.Tech. II Year II Sem.

L	T	P	C
3	0	0	3

Course Overview:

The purpose of this course is to provide a clear understanding of fundamentals with emphasis on their applications to create and manage large data sets. The course includes database design principles, normalization, concurrent transaction processing, security, recovery and file organization techniques. By using a DBMS, you can easily backup and recover the data. In addition, multi-user access to the database ensures that everyone can work together effectively.

Course Objectives: The students will try to learn

- Understand the role of database management system in an organization and learn the database concepts.
- Design databases using data modeling and Logical database design techniques.
- Construct database queries using relational algebra and calculus and SQL.
- Understand the concept of a database transaction and related concurrent, recovery facilities.
- Learn how to evaluate a set of queries in query processing

Course Outcomes: After successful completion of the course, students should be able to

- Apply knowledge of fundamentals of DBMS, Database Design
- Understand Relational Model, Relational Algebra and Calculus
- Master the basics of SQL for retrieval and management of data.
- Understand transaction processing and concurrency control.
- Use database storage structures and access techniques

Module - I **[8]**

Database Systems: A Historical Perspective, File Systems versus a DBMS, Relational Model, Levels of Abstraction in a DBMS, Data Independence, Structure of 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 ER Model.

Module - II **[9]**

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

Relational Algebra and Calculus: Relational Algebra, Tuple relational Calculus, Domain relational calculus.

Module –III **[9]**

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 data bases.

Schema refinement: Problems caused by redundancy, Decompositions, problems related to decomposition, Reasoning about Functional Dependencies, FIRST, SECOND, THIRD normal forms, BCNF, Lossless join decomposition, Multi-valued dependencies, FOURTH normal form, FIFTH normal form.

Module –IV **[10]**

Transaction Management: ACID properties, Transactions and Schedules, Concurrent execution of transactions, Lock-based Concurrency control, Performance of locking, Transaction support in SQL, Introduction to crash recovery.

Concurrency control: Serializability and Recoverability, Introduction to lock management, Lock conversions, Dealing with dead locks, Specialized locking techniques, Concurrency control without locking**.**

Module – V **[9]**

Storage and Indexing: Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree base Indexing, Comparison of File Organizations, Indexes and Performance Tuning

Tree structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure.

TEXTBOOKS:

1. Database Management Systems, Raghu Ramakrishanan, Johannes Gehrke, Tata McGraw Hill 3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGraw hill, Vth edition

REFERENCES:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel, 7th Edition.
2. SQL The Complete Reference, James R. Groff, Paul N. Weinberg, 3rd Edition,
3. Oracle for Professionals, The X Team, S.Shah and V. Shah,SPD.
4. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL,Shah,PHI.

2446606: COMPUTER SYSTEM ARCHITECTURE

B.Tech. II Year II Sem.

L	T	P	C
3	1	0	4

Course Overview:

This course introduces the concepts of Digital Computer and the basic Structure of each component. The course majorly works on Data Representation, Decoders, Multiplexers and Memory Unit, Design of Hardwired & Micro Control unit, I/O interface, Array processing units, Instruction pipeline, vector processing, multi-processor and characteristics of multi-processor.

Prerequisites:

- NIL

Course Objectives: The students will try to learn

- The functional blocks of Digital Computer, Logic gates and Flip-flops
- Digital Components, Data Representation & Error Detection Codes
- How to perform Register Transfer Micro Operations and Programming Basic Computer.
- Basic processing units and Different Memory systems
- About Pipeline and vector processing units along with I/O Organization

Course Outcomes: After successful completion of the course, students should be able to

- Understand the basic components of Digital computer.
- Demonstrate different types of Digital components and their functional units of a digital computer.
- Evaluate Different types of Micro operations on registers stored in digital computers
- Design Basic Processing units like Hardwired, Micro programmed control unit & memory.
- Design a I/O Interface for communicating with different pipelines

Module-I **[12]**

Digital Logic Circuits: Digital Computers, logic gates, Boolean algebra, map simplifications, combination circuits, Flip Flops, Sequential circuits, Flip Flop Design Procedure.

Module – II **[8]**

Digital Components: Decoders, Multiplexers, Registers, Binary Counters, Memory Unit

Data Representation: Data Types, Complements, Fixed Point representation, Floating point representation, other binary codes, Error detection codes**.**

Module–III **[9]**

Register Transfer and Micro Operations: register transfer language, register transfer, bus and memory transfer, arithmetic, logic and shift micro operations, arithmetic logic shift unit.

Programming Basic Computer: Machine Language, Assembly language, Assembler, Programming loops, Programming arithmetic and logic operations, Subroutines.

Module–IV **[9]**

Basic Processing Unit: Hardwired Control, Micro programmed Control

Memory Systems: Memory Hierarchy, Main memory, cache memories, virtual memory, secondary storage, memory management hardware.

Module–V **[9]**

Input / Output Organization: Introduction to I/O, I/O interface, Modes of transfer, Direct memory access.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor. **Multi Processors**: Characteristics of Multiprocessors

TEXTBOOKS:

1. Computer Systems Architecture – M.Moris Mano, IIIrd Edition, Pearson.

REFERENCES:

1. Computer Architecture and Organization- An Integrated Approach, Miles Murdocca, Vincent Heuring, Second Edition, WileyIndia.
2. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson
3. Computer- organization and Design- David A. Paterson and JohnL.Hennessy-Elsevier.
4. Fundamentals or Computer Organization and Design, - SivaramaDandamudi Springer Int. Edition.
5. Digital Design – Third Edition, M.Morris Mano, Pearson Education/PHI.
6. Fundamentals of Logic Design, Roth, 5th Edition, Thomson.

2440508: DESIGN AND ANALYSIS OF ALGORITHMS

B.Tech. II Year II Sem.

L	T	P	C
2	0	0	2

Course Overview:

Design and Analysis of Algorithms is a fundamental aspect of computer science that involves creating efficient solutions to computational problems and evaluating their performance. DSA focuses on designing algorithms that effectively address specific challenges and analyzing their efficiency in terms of time and space complexity.

Prerequisites:

- A course on “Data structures”.

Course Objectives: The students will try to learn

- Asymptotic performance of algorithms
- Algorithm design strategies to solve science and engineering problems.
- Concepts greedy method and dynamic programming. Applying for several applications like knapsack problem, job sequencing with deadlines, and optimal binary search tree, TSP and so on respectively.
- The methods of backtracking and branch bound techniques to solve the problems like n-queens problem, graph colouring and TSP respectively.
- Concepts of deterministic and non-deterministic algorithms

Course Outcomes: After successful completion of the course, students should be able to

- Compare asymptotic behavior of functions derived from algorithms
- Use Divide and Conquer strategy to address real time problems
- Apply greedy algorithmic design paradigm to solve problems
- Design algorithms using Dynamic Programming and backtracking strategy
- Develop algorithms for problems using branch & bound algorithm design techniques and understand NP-Hard and NP- Complete problems

UNIT-I

Module - I **[10]**

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

Disjoint Sets: Introduction, union and find Operations.

Module- II **[8]**

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen’s matrix multiplication

Module –III **[9]**

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

Dynamic Programming: General method, applications- All pairs shortest path problem, Optimal binary search trees

Module-IV **[10]**

Dynamic Programming: 0/1 knapsack problem, Reliability design ,Traveling sales person problem.

Backtracking: General method, applications, n-queen's problem, sum of subsets problem, graph coloring, Hamiltonian cycles

Module– V **[9]**

Branch and Bound: General method, applications - 0/1 knapsack problem, LC Branch and Bound solution, FIFO Branch and Bound solution, , Travelling sales person problem,

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

TEXTBOOKS:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, SatrajSahni and Raja sekharan, University Press.

REFERENCES:

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.

2440511: OOPS Through JAVA

B.Tech. II Year II – Sem.

L	T	P	C
3	0	0	3

Course Overview:

OOPs Through java makes it possible to create full reusable applications with less code and sorted development time. This course is about the fundamentals of Object-Oriented Programming (OOP) Concept and OOP-based software development methodology. It encourages modular objects for reusable code, ensure well organize and maintainable code via encapsulation, inheritance and polymorphism. OOP finds broad application in software development domains:

- Software Development
- GUI Development
- Game Development
- Database Systems
- Simulation and Modeling

Prerequisites:

- A course on Problem Solving Using C and C++

Course Objectives: The students will try to learn

- Concepts and features of object oriented programming
- Java Standard API library such as util, io, applets, GUI based controls.
- Exception handling mechanism, multithreading, packages and interfaces.
- How to use Collection framework
- Internet programming using applets and AWT.

Course Outcomes: After successful completion of the course, students should be able to

- Solve real world problems using OOP concepts.
- Understand the use of abstract classes and Interfaces
- Develop multithreaded applications with synchronization.
- Solve problems using java collection framework
- Develop applications using Event Handling

Module - I **[10]**

Principles of OOPS: OOPS Paradigm, Objects, Classes and Methods, Abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic Binding.

JAVA Introduction: History of Java, Java buzzwords, data types, variables, scope and life time of variables, Type conversion and casting, arrays, operators, Operator Precedence, control statements.

Java String Handling: String Constructors, Special string operations, Character Extraction, String Comparisons, Modifying a string, String Buffer.

Module - II **[9]**

Classes: Class fundamentals, Declaring Objects, methods, Constructors, this keyword, garbage collection, Overloading methods and constructors, Recursion.

Inheritance, Packages and Interfaces – Inheritance basics, Using super, Creating a multilevel hierarchy, method overriding, Dynamic method dispatch, abstract classes, Using final with inheritance, Defining a package, Finding package and class path, Access protection, importing packages, Defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces**.**

Module –III **[9]**

Exception handling - Exception types, uncaught exceptions, using try and catch, Multiple catch classes, nested try statements, throw, throws and finally. Java's built-in exceptions, creating own exception sub classes.

Multithreading - Thread life cycle, Thread Creation using thread class and runnable interface, Creating multiple threads, Thread priorities, Synchronizing threads, Inter thread communication.

Module –IV **[9]**

Collections Framework: Overview, Collection Interfaces, Collection Classes, Accessing a collection via Iterator, Working with Maps

Java Database Connectivity: Types of Drivers, JDBC architecture, JDBC Classes and Interfaces, Basic steps in Developing JDBC Application, Creating a New Database and Table with JDBC.

Module– V **[9]**

GUI Programming with Swing – Introduction, limitations of AWT, MVC architecture, components, containers, Layout Manager Classes, Simple Applications using AWT and Swing.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes.

TEXT BOOKS:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.(Principles of OOPS in UNIT-I)

REFERENCES:

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.

3. Object Oriented Programming through Java, P. Radha Krishna, University Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.
5. Java Programming and Object-oriented Application Development, R. A. Johnson, Cengage Learning

2440581: Artificial Intelligence Laboratory

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

0 0 2 1

Course Outcomes: The students will be able to:

- Understand current of advanced AI technologies that enable machines to sense, comprehend, actand learn on their own.
- Distinguish class of problems suitable for solving with AI.
- Design, implement and evaluate a computing-based solution to meet a given set of computing requirements in the context of the AI.
- Design and create AI suitable for solving particular problem.
- Apply basic principles of AI in solutions that require problem solving, inference, perception,knowledge representation, and learning.

LIST OF EXPERIMENTS:

1. Write a LISP code to perform Arithmetic operations.
2. Write a Recursive LISP function which takes one argument as a list and return reverse ofthe list.
3. Write a LISP function to compute difference of squares.(if x > y return x2– y2 , Otherwisey2 –x2).
4. Write simple fact for following:
   A. Ram likes mango.
   B. Seema is a girl.
   C. Bill likes Cindy.
   D. Rose is red.
   E. John owns gold
5. Write a prolog program that convert temperature from Celsius to Fahrenheit
6. Write simple Prolog functions such as the following. Take into account lists which are tooshort. --remove the Nth item from the list. -- insert as the Nth item.
7. Write a Program to Implement Tic-Tac-Toe game.
8. Write a Program to Implement 8-Puzzle problem
9. Write a Program to Implement Water-Jug problem
10. Write a Program to Implement Monkey Banana Problem.
11. Write a Program to Implement N-Queens Problem.
12. Write a Program to Implement Min-Max Algorithm.
13. Implementation of TSP using heuristic approach using Prolog
14. Implementation of Simulated Annealing Algorithm using PROLOG
15. Implementation of Hill-climbing to solve 8- Puzzle Problem
16. Write a Program see all elements are present in the list or not.
17. Write a Program to login so that user can attempt 3 times. After 3rd attempt program mustterminate with message “NOT PERMITTED FOR LOGIN”

TEXTBOOKS:

1. Artificial Intelligence A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson Education

REFERENCES:

1. Artificial Intelligence, E.Rich and K.Knight, , 3rd Edition, TMH
2. Artificial Intelligence, Patrick Henny Winston, 3rd Edition, Pearson Education.
3. Artificial Intelligence, ShivaniGoel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

2440573: DATABASE MANAGEMENT SYSTEMS LAB

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

Course Overview:

The purpose of this course is to provide a clear understanding of fundamentals with emphasis on their applications to create and manage large data sets. The course includes database design principles, normalization, concurrent transaction processing, security, recovery and file organization techniques. By using a DBMS, you can easily backup and recover the data. In addition, multi-user access to the database ensures that everyone can work together effectively.

Course Objectives: The students will try to learn

- Developing of ER diagrams for the given problem.
- Design databases using data modeling and Logical database design techniques.
- Construction of database queries using relational algebra and calculus and SQL.
- The concept of a database transaction and related concurrent, recovery facilities.
- Creation of triggers, stored procedures and cursors.

Course Outcomes: After successful completion of the course, students should be able to

- Develop ER diagram for the given problem.
- Understand Relational Model, Relational Algebra and Calculus
- Master the basics of SQL for retrieval and management of data.
- Creation of triggers
- Use stored procedures and cursors.

Problem statement

"Roadway Travels" is in business since 1997 with several buses connecting different places in india. Its main office is located in Hyderabad.

The company wants to computerize its operations in the following areas:

- Reservation and Ticketing
- Cancellations

Reservation & Cancellation:

Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family).

Cancellations are also directly handed at the booking office.

In the process of computerization of Roadway Travels you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query's using SQL to retrieve the data from database.

The above process involves many steps like
1. Analyzing the problem and identifying the Entities and Relationships,
2. E-R Model
3. Relational Model
4. Normalization
5. Creating the database
6. Querying.

Students are supposed to work on these steps week wise and finally create a complete "Database System" to Roadway Travels. Examples are given at every experiment for guidance to students.

Experiment 1: E-R Model

Analyze the carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the others keys like candidate keys, partial keys, if any.

Example: Entities:
1. BUS
2. Ticket
3. Passenger

Relationships:
1. Reservation
2. Cancellation

PRIMARY KEY ATTRIBUTES:
Ticket ID (Ticket Entity)
Passport ID (Passenger Entity)
Bus_No (Bus Entity)

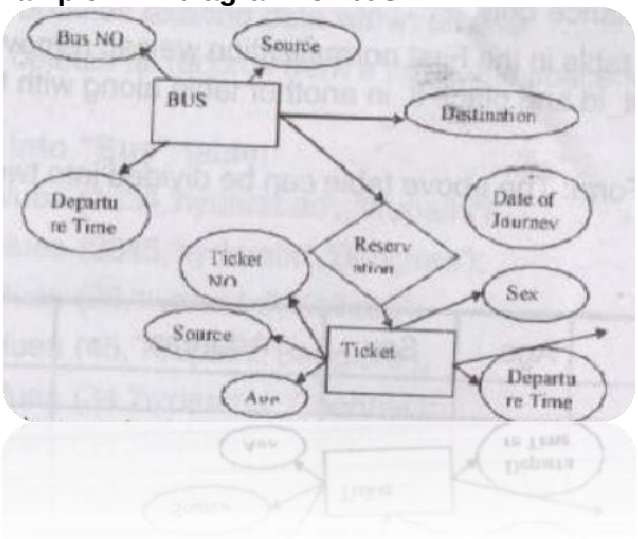
A part from the above mentioned entities you can identify more. The above mentioned are few.

Note: The students is required to submit a document by writing the Entities and keys to the lab teacher.

Experiment 2: Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

Example: E-R diagram for bus



Note: The students is required to submit a document by drawing the E-R Diagram.

Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. Thereare different ways of representing relationships as tables based on the requirement. Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

Example: The passenger tables look as below. This is an example. You can add more attributes based on E-Rmodel. This is not a normalized table.

Passenger

Name	Age	Sex	Address	Ticket_id	Passport ID

Note: The students is required to submit a document by Represent relationships in a tabular fashion to the labteacher.

Experiment 4: Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only. For the above table in the First normalization we can remove the multiple valued attribute Ticket_id and placeit in another table along with the primary key of passenger.

First Normal Form: The above table can divided into two tables as shown below.Passenger

Name	Age	Sex	Address	Passport ID

Passport ID	**Ticket_id**

You can do the second and third normal forms if required. Any how Normalized tables are given at the end.

Experiment 5: Installation of MySQL and practice DDL commands

Installation of MySQL. In this week you will learn Creating databases, How to create tables, altering the database, droping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized "Passenger"
table.CREATE TABLE Passenger(
Passport_id INTEGER PRIMARY KEY,
Name VARCHAR(50) NOT NULL, Age
INTEGER NOT NULL,
Sex CHAR,
Address VARCHAR(50) NOT NULL
);
Similarly create all other tables.
Note: Detailed creation of tables is given at the end.

Experiment 6: Practicing DML commands

DML commandsare used for managing data within schema objects. Some examples:

- SELECT - retrieve data from the database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

insert values into "Bus" table:
insert into Bus values (1234, 'hyderabad', 'tirupathi');

insert values into "Passenger" table:
insert into Passenger values(1, 45, 'ramesh', 45, 'M',
'abc123');insert into Passenger values(56, 22, 'seetha', 32,
'F', 'abc55');

Few more Examples of DML commands:
SELECT * FROM Bus; (selects all the attributes and
displays)UPDATE Bus SET Bus_No = 1 WHERE Bus_No =
2;

Experiment 7: Querying

In this week you are going to practice queries(along with sub queries) using ANY, ALL, IN, EXISTS, NOT EXIST, UNION, INTERSECT, Constraints etc.

Practice the following Queries:

Display unique PNR_No of all passengers.
Display all the names of male passengers.
Display the ticket numbers and names of all the passengers.
Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.
Find the names of passengers whose age is between 30 and 45.
Display all the passengers names beginning with 'A'
Display the sorted list of passengers names.

Experiment 8 and Experiment 9: Querying (continued...)
You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, MAX,and MIN), GROUP BY, HAVING and Creation and dropping of VIEWS.

Write a Query to display the information present in the Passenger and cancellation tables. Hint: UseUNION Operator.
Display the number of days in a week on which the 9W01 bus is available.
Find number of tickets booked for each PNR_No using GROUP BY CLAUSE. Hint: Use GROUP BY on PNR_No.
Find the distinct PNR numbers that are present.
Find the number of tickets booked by a passenger where the number of seats is greater than 1. Hint:Use GROUP BY, WHERE and HAVING CLAUSES.
Find the total number of cancelled seats.

Experiment 10: Triggers
. Creation of insert trigger, delete trigger, update trigger.Practice triggers using the above database.
E.g:

```
CREATE TRIGGER updatecheck BEFORE UPDATE ON passenger FOR EACH ROW BEGIN
IF NEW.TickentNO> 60 THEN
SET New.TickentNO = TicketNo; ELSE
SET New.TicketNo = 0;END IF;
END
```

Experiment 11; Procedures
Creation of stored procedure, Execution of procedure and modificationof procedure. Practice procedures using the database.
E.g:

```
CREATE PROCEDURE myproc() BEGIN
SELECT COUNT(Tickets) FROM Ticket
WHERE age >= 40; END;
```

Experiment 12: Cursors
In this week you need to do the following: Declare a cursor that defines a result set.
Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, onerow at a time. Close the cursor when done

```
CREATE PROCEDURE myproc(in_customer_id INT)
BEGIN
  DECLARE v_id INT;
  DECLARE v_nameVARCHAR(30);

  DECLARE c1 CURSOR FOR
  SELECT stdid, stdFirstname FROM studentsss WHERE stdid - in_customer_id;

  OPEN c1;
  FETCH c1 INTO v_id, v_name; CLOSE c1;
  END;
```

Tables:
BUS
Bus No: VARCAHR : PK(primary key) Source: VARCHAR
Destination: VARCHAR

Passenger
PPNO: VARCHAR(15) : PK Name: VARCHAR(15)
Age: INT(4)
Sex: CHAR(10) : Male/FemaleAddress: VARCHAR(20) **Passenger_Tickets**
PPNO: VARCHAR(15) : PK
Ticket_No: NUMERIC(9)
Reservation
PNR_No: NUMERIC(9) : FK
Journey_date: DATETIME(8) No_of_seats: INT(8) Address: VARCHRA(50)
Contact_No: NUMERIC(9) --> Should not less than 9 and Should not accept any other character other than interger
STATUS: CHAR(2) : Yes/No
Cancellation
PNR_No: NUMERIC(9) : FK
Journey_date: DATETIME(8) No_of_seats: INT(8) Address: VARCHRA(50)
Contact_No: NUMERIC(9) --> Should not less than 9 and Should not accept any other character other than interger
STATUS: CHAR(2) : Yes/No
Ticket
Ticket_No: NUMERIC(9) : FK Journey_date: DATETIME(8) Age: INT(4)
Sex: CHAR(10) : Male/Female Source: VARCHAR Destination: VARCHAR Dep_time: VARCHAR

REFERENCES:

1. Introduction to SQL, Rick F.vanderLans, Pearson education.
2. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, Pearson education.
3. Oracle PL/SQL Programming, Steven Feuerstein, SPD.
4. SQL & PL/SQL for Oracle 10g, Black Book, Dr. P.S. Deshpande, Dream Tech.
5. Oracle Database 11g PL/SQL Programming, M. Mc Laughlin, TMH.
6. SQL Fundamentals, J.J. Patrick, Pearson Education

2440573:DESIGN AND ANALYSIS OF ALGORITHMS THROUGH JAVA LABORATORY

B.Tech. II Year II Sem.

L	T	P	C
0	0	2	1

Course overview:
This course focuses on problem solving using Object oriented concepts by using JAVA programming. It also gives insight implementation of algorithms using JAVA.

Prerequisites: Problem Solving Using C and C++

Co-requisites : Design and Analysis of Algorithms

Course Objectives: The students will try to learn

- Installing and using of any IDE
- Java Standard API library such as util, io, applets, GUI based controls.
- Exception handling mechanism, multithreading, packages and interfaces.
- How to use Collection framework
- How to implement different algorithm models using java..

Course Outcomes: After successful completion of the course, students should be able to

- Install and use any IDE.
- Understand the use of abstract classes and Interfaces
- Develop multithreaded applications with synchronization.
- Solve problems using java collection framework
- Implement algorithms using JAVA.

List of Programs:

1.

a. Use Eclipse or Net bean platform and acquaint 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.

b. Write a java program that prints all real solutions to the quadratic equation ax2 +bx+c=0. Read in a, b, c and use the quadratic formula.

c. Write a java program to implement Fibonacci series.

1. Write Java Programs to perform following:

a. To count occurrence of each character in a string.

b. To remove duplicate words from a string.

c. To print all permutations of a string.

3.

a. Write a java program to implement method overloading and constructors overloading.

b. Write a java program to implement method overriding.

c. 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.

4.

a. Write a java program to check whether a given string is palindrome.

b. 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.

5.

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

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

6. Write programs to implement following using Collection Framework:

a. to add, retrieve & remove element from ArrayList
b. to Sort & reverse the LinkedList elements
c. to sort ArrayList using Comparable and Comparator

7. Write programs to implement following using Collection Framework:

a. to copy elements from HashSet to Array
b. to remove duplicate key from hashtable
c. to iterate TreeMap

8. Write a program to implement Knapsack problem using greedy method.

9. Write a program to implement Job sequencing with deadlines and Single source shortest path problem using Greedy Method

10. Write a program to implement All pairs Shortest path and 0/1 Knapsack problem using Dynamic Programming

11. Write a program to implement Optimal Binary Search Tree using Dynamic Programming

12. Write a program to implement n-Queen‟s problem and Sum of subsets using backtracking method.

13. Write a program to implement Travelling sales person using branch and bound, dynamic programming

TEXT BOOKS:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, SatrajSahni 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./ PearsonEducation.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley andsons
4. Java The Complete Reference, Herbert Schildt‟s, 9th Edition, TATA McGRAW –HILL.

2440597: NoSQL Data Bases (MongoDB)

B.Tech. II Year II Sem.

L	T	P	C
0	0	2	1

Programs can be implemented using the **MongoDB Shell** or with **Python/Java/PHP** for connectivity where appropriate.

Module I: Introduction & Data Modeling

1. **Create and Explore a NoSQL Document Structure**
   - Insert sample JSON documents to show flexible schemas.
2. **Compare RDBMS vs. MongoDB with a Practical Schema**
   - Model the same data (e.g., user accounts) in SQL and MongoDB.
3. **Explore MongoDB Data Types**
   - Insert and query documents using types like `String`, `NumberInt`, `Boolean`, `Array`, `Date`, etc.
4. **Basic MongoDB Data Modeling Example**
   - Design an embedded vs. referenced model for blog posts and comments.

Module II: Operators and Commands

1. **Use Query and Projection Operators**
   - Demonstrate `$eq`, `$gt`, `$lt`, `$in`, `$and`, `$or`, `$exists`, and projection `{ field: 1 }`.
2. **Update Operators and Aggregation Stages**
   - Use `$set`, `$inc`, `$push` in `update`, and pipeline stages like `$match`, `$group`, `$sort`.
3. **Sorting, Limiting, and Modifying Queries**
   - Apply `.limit()`, `.sort()`, and modifiers like `.explain()`, `.hint()`.
4. **Geospatial Commands and User Management**
   - Insert geoJSON data and run `$geoWithin` queries; create users and assign roles.

Module III: Database & Collection Management

1. **Create and Drop a Database**
   - Use `use dbName`, `db.dropDatabase()`.
2. **Create and Drop Collections**
   - `db.createCollection("students")`, `db.students.drop()`.

3. **Explore Collection Indexes and Options**
    - Create indexes and check with `db.collection.getIndexes()`.
4. **Set Up Schema Validation Rules**
    - Use JSON schema validation to restrict document structure.

Module IV: CRUD Operations & System Commands

1. **CRUD: Insert, Query, Update, Delete Documents**
    - Full example of inserting, querying with filters, updating fields, and deleting.
2. **Use of `db.runCommand()` and Server Info**
    - Run `db.runCommand({ serverStatus: 1 })` and `db.isMaster()`.
3. **Bulk Operations and Upsert Example**
    - Demonstrate `bulkWrite()` with mixed inserts and updates.
4. **Check Collection Stats and Perform Partial Updates**
    - `db.collection.stats()`, `$set` with field targeting.

Module V: Shell, Methods, and Connectivity

1. **Using MongoDB Shell: Collection and Cursor Methods**
    - Demonstrate `.find()`, `.countDocuments()`, `.forEach()`, `.toArray()`.
2. **Query Plan Cache and Role Management Commands**
    - `db.collection.getPlanCache().clear()` and role creation with `db.createRole()`.
3. **Python MongoDB CRUD App using PyMongo**
    - Connect to MongoDB Atlas/local, and perform CRUD using Python.

III YEAR I SEMESTER (V SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2450513	Computer Networks	PC	3	1	0	4	40	60	100
2	2450504	Formal Languages and Automata Theory	PC	3	0	0	3	40	60	100
3	2450515	Machine Learning	PC	3	0	0	3	40	60	100
4		Professional Elective-I	PE	3	0	0	3	40	60	100
5		Open Elective- I	OE	3	0	0	3	40	60	100
		Laboratory								
1	2450579	Computer Networks Laboratory	PC	0	0	2	1	40	60	100
2	2450585	Devops Laboratory	PC	0	0	2	1	40	60	100
3	2450583	Machine Learning Laboratory	PC	0	0	2	1	40	60	100
		Project								
1	2450587	Field Based Project	PS	0	0	2	1	100	-	100
		Mandatory Course								
1	2450022	Gender Sensitization	MC	0	0	0	0	-	-	-
Total Credits				**15**	**1**	**8**	**20**	**420**	**480**	**900**

2450513: COMPUTER NETWORKS

L T P C
3 1 0 4

Prerequisites: Data Structures

Course Objectives:

- To introduce the basic concepts of data communications, transmission media, switching techniques, and network reference models (OSI & TCP/IP).
- To develop an understanding of data link layer functions such as framing, error detection and correction, flow/error control, and medium access protocols used in wired and wireless networks.
- To explain the principles of network layer operations, including logical addressing, routing algorithms, address mapping, and congestion control mechanisms.
- To impart knowledge of transport layer services and protocols with emphasis on reliable communication, TCP/UDP mechanisms, congestion handling, and quality of service (QoS).
- To familiarize students with application layer protocols and enabling effective communication over the Internet.

Course Outcomes: After completion of the course, students should be able to

- Explain the fundamentals of data communication, network structures, transmission media, switching techniques, and reference models (OSI & TCP/IP).
- Apply error detection/correction techniques and medium access control protocols for ensuring reliable and efficient data link layer communication.
- Analyze logical addressing, routing algorithms, and congestion control techniques for effective network layer operations.
- Evaluate transport layer protocols (TCP/UDP) and mechanisms for process-to-process delivery, congestion control, and Quality of Service (QoS).
- Demonstrate the use of application layer protocols such as DNS, SMTP, FTP, HTTP, and SNMP in real-world networking scenarios.

MODULE – I [10]

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.

Module – II [10]

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

Module – III [10]

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.

Module – IV [8]

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

Module – V [10]

Application Layer: Principles of Network Applications, Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, SMTP, DNS, Peer-to-Peer Applications, Socket Programming: Creating Network Applications.

TEXT BOOK:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI
2. Computer Networking: A Top-Down Approach - James F.Kurose, Keith W. Ross, Pearson

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.

2450504: Formal Languages and Automata Theory

L	T	P	C
3	0	0	3

Prerequisites: Discrete Mathematics, Data structures

Course Objectives:

- To introduce the foundational concepts of automata theory, formal languages, and their role in computation.
- To develop the ability to design finite automata, regular expressions, and grammars for language representation.
- To familiarize students with context-free grammars, pushdown automata, and their applications in parsing.
- To provide knowledge of normal forms, closure properties, and pumping lemmas for analyzing languages.
- To explain Turing machines, undecidability, and computational limits of formal models.

Course Outcomes: After completion of the course, students should be able to

- Apply the fundamental concepts of finite automata, regular languages, and their conversions for solving computational problems.
- Construct and analyze regular expressions, automata, and grammars for representing formal languages.
- Design context-free grammars and pushdown automata for language recognition and parsing applications.
- Demonstrate the use of normal forms, closure properties, and pumping lemmas for proving properties of languages.
- Evaluate the computational power of Turing machines and analyze decidability and undecidability problems in automata theory.

Module-I **[10]**

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems. Non deterministic 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.

Module-II **[10]**

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.

Module-III **[10]**

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Tress, 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. Conversion of CFG to PDA.

Module-IV **[10]**

Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €- Productions. Chomsky Normal form Griebech 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.

Module-V **[8]**

Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine 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.

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 Chandra shekaran, 2nd edition, PHI.

REFERENCES:

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 Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press

2450515: MACHINE LEARNING

L T P C
3 0 0 3

Prerequisite: Mathematical Statistical Foundation, Data Structures

Course Objectives:

- To introduce the fundamental concepts, types, and issues of machine learning, along with the theoretical foundations of supervised and unsupervised learning.
- To develop the ability to design and implement learning systems using models such as perceptrons, multilayer neural networks, decision trees, and support vector machines.
- To explore advanced machine learning techniques including ensemble methods, radial basis functions, and dimensionality reduction approaches.
- To provide insights into evolutionary and probabilistic learning models such as genetic algorithms, Bayesian networks, and hidden Markov models.
- To enable students to apply reinforcement learning and modern optimization strategies for solving real-world machine learning problems effectively.

Course Outcomes: After completion of the course, student should be able to

- Distinguish between, supervised, unsupervised and semi-supervised learning
- Describe the relationship between the brain, neuron models, and learning systems, highlighting their computational perspectives.
- Formulate concept learning tasks and apply hypothesis search techniques such as maximally specific hypothesis and version spaces.
- Implement linear models including linear discriminants, perceptron learning, and linear regression for classification and prediction.
- Evaluate issues and perspectives in machine learning system design by considering challenges such as linear separability and algorithm limitations.

Module – I [10]

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, Logistic Regression.

Module – II [10]

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

Module – III [8]

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

Evaluation Metrics in Machine Learning: Confusion Matrix, Accuracy, Precision, Recall, F1 score

Module – IV [10]

Dimensionality Reduction: Linear Discriminant Analysis, Principal Component Analysis, Factor Analysis, Independent Component Analysis, Locally Linear Embedding, Least Squares Optimization

Evolutionary Learning: Genetic algorithms – Genetic Offspring: - Genetic Operators – Using Genetic Algorithms

Module – V [10]

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

TEXT BOOKS:

1. Stephen Marsland, —Machine Learning — An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series.

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

24X6641: ETHICAL HACKING

L T P C
3 0 0 3

Prerequisites: Operating Systems, Computer Networks

Course Objectives:

- To introduce the methodologies and framework of ethical hacking for enhancing security.
- To understand Impacts of Hacking; Types of Hackers; Information Security Models.
- To understand the methodologies, framework, and impacts of ethical hacking
- To Plan and execute a controlled, ethically-bounded security assessment by applying the hacker framework
- To explore Information Security Program, Business Perspective, Planning a Controlled Attack.

Course Outcomes: After completion of the course, students should be able to

- Understand availability of tools for supporting an ethical hack
- Gain the knowledge of interpreting the results of a controlled attack
- Explore the role of politics, inherent and imposed limitations and metrics for planning of a test
- Comprehend the dangers associated with penetration testing
- Differentiate among computer, network, service, application and architecture security models

MODULE – I [10]

Introduction: Hacking Impacts, The Hacker Framework: Planning the test, Sound Operations, Reconnaissance, Enumeration, Vulnerability Analysis, Exploitation, Final Analysis, Deliverable, Integration

Information Security Models: Computer Security, Network Security, Service Security, Application Security, Security Architecture Information Security Program: The Process of Information Security, Component Parts of Information Security Program, Risk Analysis and Ethical Hacking

MODULE – II [10]

The Business Perspective: Business Objectives, Security Policy, Previous Test Results, Business Challenges Planning for a Controlled Attack: Inherent Limitations, Imposed Limitations, timing is Everything, Attack Type, Source Point, Required Knowledge, Multi-Phased Attacks, Teaming and Attack Structure, Engagement Planner, The Right Security Consultant, The Tester, Logistics, Intermediates, Law Enforcement

MODULE – III [10]

Preparing for a Hack: Technical Preparation, Managing the Engagement Reconnaissance: Social Engineering, Physical Security, Internet Reconnaissance

MODULE – IV [10]

Enumeration: Enumeration Techniques, Soft Objective, Looking Around or Attack, Elements of Enumeration, Preparing for the Next Phase Exploitation: Intuitive Testing, Evasion, Threads and Groups, Operating Systems, Password Crackers, RootKits, applications, Wardialing, Network, Services and Areas of Concern

MODULE – V [8]

Deliverable: The Deliverable, The Document, Overall Structure, Aligning Findings, Presentation Integration: Integrating the Results, Integration Summary, Mitigation, Defense Planning, Incident Management, Security Policy, Conclusion

TEXT BOOK:

1. James S. Tiller, “The Ethical Hack: A Framework for Business Value Penetration Testing”, Auerbach Publications, CRC Press.

REFERENCE BOOKS:

1. EC-Council, “Ethical Hacking and Countermeasures Attack Phases”, Cengage Learning.

2. Michael Simpson, Kent Backman, James Corley, “Hands-On Ethical Hacking and Network Defense”, Cengage Learning

24X0529: DATA SCIENCE

L T P C
3 0 0 3

Course Objectives:

- To Learn concepts, techniques and tools they need to deal with various facets of data science practice, including data collection and integration
- Understand the basic types of data and basic statistics
- Identify the importance of data reduction and data visualization techniques
- Introduce the fundamental concepts of Data Science, including big data, datafication, and statistical inference.
- Develop practical skills in R programming for handling vectors, matrices, factors, data frames, and lists.

Course Outcomes: After completion of the course, student should be able to

- Understand basic terms of statistical modeling and data science
- Implement of R programming concepts
- Utilize R elements for data visualization and prediction
- Differentiate between various data types and attributes, and summarize data using statistical measures and graphical representations.
- Construct data visualizations (charts, plots, histograms, scatterplots) and perform regression analysis (linear and multiple regression) for interpreting and modelling of data.

MODULE – I [10]

Definition of Data Science- Big Data and Data Science hype – and getting past the hype - Datafication- Current landscape of perspectives - Statistical Inference - Populations and samples – Statistical modeling, probability distributions, fitting a model – Over fitting. Basics of R: Introduction, R-Environment Setup, Programming with R, Basic Data Types.

MODULE – II [10]

Data Types & Statistical Description: Types of Data: Attributes and Measurement, Attribute, The Type of an Attribute, The Different Types of Attributes, Describing Attributes by the Number of Values, Asymmetric Attributes, Binary Attribute, Nominal Attributes, Ordinal Attributes,

Numeric Attributes, Discrete versus Continuous Attributes. Basic Statistical Descriptions of Data: Measuring the Central Tendency: Mean, Median, and Mode, Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range, Graphic Displays of Basic Statistical Descriptions of Data.

MODULE – III [10]

Vectors: Creating and Naming Vectors, Vector Arithmetic, Vector sub setting, Matrices: Creating and Naming Matrices, Matrix Sub setting, Arrays, Class. Factors and Data Frames: Introduction to Factors: Factor Levels, Summarizing a Factor, Ordered Factors, Comparing Ordered Factors, Introduction to Data Frame, subsetting of Data Frames, Extending Data Frames, Sorting Data Frames. Lists: Introduction, creating a List: Creating a Named List, Accessing List Elements, Manipulating List Elements, Merging Lists, Converting Lists to Vectors

MODULE – IV [10]

Conditionals and Control Flow: Relational Operators, Relational Operators and Vectors, Logical Operators, Logical Operators and Vectors, Conditional Statements. Iterative Programming in R: Introduction, While Loop, For Loop, Looping Over List. Functions in R: Introduction, writing a Function in R, Nested Functions, Function Scoping, Recursion, Loading an R Package, Mathematical Functions in R.

MODULE – V [8]

Charts and Graphs: Introduction, Pie Chart: Chart Legend, Bar Chart, custom charts, cyclical data and circular area charts, Dual Axis charts, Box Plot, Histogram, Line Graph: Multiple Lines in Line Graph, Scatter Plot.

Regression: Linear Regression Analysis, Multiple Linear regression

TEXT BOOKS:

1. Doing Data Science, Straight Talk from The Frontline. Cathy O'Neil and Rachel Schutt, O'Reilly,2014.

2. K G Srinivas, G M Siddesh, "Statistical programming in R", Oxford Publications.

REFERENCE BOOKS:

1. Jiawei Han, Micheline Kamber and Jian Pei. Data Mining: Concepts and Techniques, 3rd ed.The Morgan Kaufmann Series in Data Management Systems.

24X6604: ARTIFICIAL NEURAL NETWORKS

L T P C
3 0 0 3

Course Objectives: The students will try to learn

- Biological and mathematical foundations of neural networks.
- Supervised learning methods such as Perceptron and Backpropagation.
- Unsupervised learning techniques including Kohonen maps and ART networks.
- Associative memory models and their implementations.
- Applications of neural networks in real-world problems.

Course Outcomes: After completion of the course, students should be able to

- Model artificial neurons and design neural network architectures.
- Train feedforward networks using Backpropagation and other algorithms.
- Implement unsupervised learning and clustering using Kohonen networks.
- Analyze associative memory networks like Hopfield and BAM.
- Apply neural networks to solve pattern recognition, classification, and optimization tasks.

Module-I **[10]**

Introduction: Biological neuron and artificial neuron models, activation functions (step, sigmoid, tanh)

Network architectures: single-layer and multi-layer feedforward networks. Perceptron: Structure, algorithm, convergence theorem, limitations of single-layer perceptron.

Module-II **[8]**

Supervised Learning: Gradient descent and Delta rule. Backpropagation algorithm: derivation, network training, convergence, problems like local minima, learning rate, momentum, overfitting, and generalization.

Module-III **[10]**

Radial Basis Function Networks: Structure and learning. Cover's theorem and its significance. Support Vector Machines: Basic concepts and kernel trick.

Unsupervised Learning: Hebbian learning rule, competitive learning, Kohonen's Self Organizing Feature Maps (SOFM): architecture, algorithm, and applications.

Module-IV **[10]**

Associative Memory: Concepts of auto-associative and hetero-associative memories.

Hopfield Networks: Architecture, energy function, learning and recall, capacity.

Bidirectional Associative Memory (BAM): Architecture and learning algorithm.

Adaptive Resonance Theory (ART): ART1 and ART2 networks, stability-plasticity dilemma.

Module-V **[10]**

Applications of Neural Networks: Function approximation and regression, Pattern recognition and classification: handwritten character recognition, speech recognition. Time series prediction. Neural network approaches for optimization: Traveling Salesman Problem (TSP), scheduling. Case studies and emerging trends in deep learning.

TEXTBOOKS:

1. Simon Haykin, “Neural Networks and Learning Machines,” 3rd Edition, Pearson Education.
2. B. Yegnanarayana, “Artificial Neural Networks,” PHI.

REFERENCES:

1. S. Rajasekaran and G.A. Vijayalakshmi Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithms,” PHI.
2. Satish Kumar, “Neural Networks: A Classroom Approach,” McGraw Hill.
3. Jacek M. Zurada, “Introduction to Artificial Neural Systems,” Jaico Publishing House.

24X0523: INTERNET OF THINGS

L T P C
3 0 0 3

Pre-Requisites: Computer organization, Computer Networks

Course Objectives:

- To introduce the fundamental concepts, architecture, and enabling technologies of the Internet of Things (IoT).
- To familiarize students with domain-specific IoT applications in areas like home automation, healthcare, agriculture, and environment.
- To explain communication protocols, system management tools, and standards essential for IoT and M2M (Machine-to-Machine) interactions.
- To develop students' ability to design and implement IoT systems using Python and physical devices such as Raspberry Pi.
- To expose students to real-world IoT case studies and cloud-based IoT platforms for data storage, visualization, and remote access.

Course Outcomes: After completion of the course, students should be able to

- Explain physical and logical design of IoT systems, and identify suitable enabling technologies.
- Differentiate between IoT and M2M communication.
- Develop IoT system components using Python, including handling data types, files, modules, and relevant packages.
- Interface IoT devices like Raspberry Pi and integrate them with cloud services for IoT-based applications.
- Analyze IoT solutions across various domains such as smart homes, environment monitoring, and agriculture.

MODULE – I [10]

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

MODULE – II [10]

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

MODULE – III [10]

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

MODULE – IV [10]

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

MODULE – V [8]

Case studies- Home Automation, Environment-weather monitoring-weather reporting- air pollution monitoring, Agriculture.

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.

2450579: COMPUTER NETWORKS LABORATORY

L T P C
0 0 2 1

Prerequisite: Data Structures Laboratory, Operating Systems Laboratory

Course Objectives:

- To Understand core data link and network layer concepts including framing, error detection (CRC), flow control, and routing protocols to build a strong theoretical foundation.
- To Implement practical network algorithms and protocols—such as character/bit stuffing, CRC (CRC-12/CRC-16/CRC-CCITT), sliding-window (Go-Back-N), Dijkstra's shortest path, and distance-vector routing—through hands-on programming assignments.
- To Develop and evaluate network mechanisms for reliability and performance, including loss recovery, congestion control (leaky bucket), buffer/frame sorting techniques, and encryption/decryption for secure data transmission.
- To Use industry tools and simulators (Wireshark, Nmap, NS2) to capture/analyze traffic, perform scans and OS detection, simulate packet drops, throughput, congestion, and compare TCP/UDP behavior under varied network conditions.
- To Analyze and interpret results from experiments and simulations to produce meaningful performance metrics (packet drop, throughput, congestion plots, broadcast trees) and to propose improvements or configuration choices based on empirical evidence.

Course Outcomes: After completion of the course, students should be able to

- Implement data link layer techniques for framing, error detection, and flow control.
- Apply routing algorithms for compute paths, routing tables, and broadcast trees.
- Develop congestion control, buffer management, and loss recovery mechanisms.
- Use Wireshark, Nmap, and NS2 tools to capture, analyze, and simulate network performance.
- Analyze results for evaluate network efficiency, reliability, and security.

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. Implement the following using Wireshark
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.
11. How to run Nmap scan Operating System Detection using Nmap

Do the following using NS2 Simulator

12 . Simulate to Find the Number of Packets Dropped

13. Simulate to Find the Number of Packets Dropped by TCP/UDP

14. Simulate to Find the Number of Packets Dropped due to Congestion

15. Simulate to Compare Data Rate & Throughput.

16. Simulate to Plot Congestion for Different Source/Destination

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

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.

2450585: DEVOPS LABORATORY

L T P C
0 0 2 1

Course Objectives:

- To introduce students to fundamental DevOps practices such as version control, continuous integration, continuous deployment, and automation.
- To enable learners to explore and apply modern tools like Git, GitHub, Jenkins, Docker, Kubernetes, and Selenium in real-world scenarios.
- To develop the ability to design and manage containerized applications, ensuring scalability, efficiency, and reliability.
- To provide hands-on experience in automating software development workflows, from source code management to deployment and testing.
- To cultivate problem-solving and analytical skills by integrating DevOps tools for building, deploying, and testing applications in a collaborative environment.

Course Outcomes: After completion of the course, student should be able to

- Apply version control tools (Git and GitHub) to manage, track, and collaborate on source code effectively.
- Configure Continuous Integration/Continuous Deployment (CI/CD) pipelines using Jenkins, ensuring automated builds and deployments.
- Develop, containerize, and deploy applications using Docker and Kubernetes**,** achieving scalability and automation in application management.
- Perform automated testing of applications using Selenium**,** integrating it with CI/CD workflows for quality assurance.
- Design and execute an end-to-end DevOps workflow, from code development and version control to containerization, deployment, and testing of applications.

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.
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.
13. Develop Package and deploy the multi-service registration app to Kubernetes using Helm charts and expose it via Ingress.
14. Convert the single registration app into a small multi-service system and orchestrate locally with Docker Compose.

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.

2450583: MACHINE LEARNING LABORATORY

L	T	P	C
0	0	2	1

Prerequisites: Essentials of Problem solving using python Laboratory

Course Objectives:

- To impart knowledge of statistical concepts and their implementation in Python for data analysis.
- To familiarize students with Python libraries such as Statistics, Math, NumPy, SciPy, Pandas, and Matplotlib for data processing and visualization.
- To develop skills in applying supervised learning techniques like Linear Regression, Logistic Regression, Decision Trees, and KNN using sklearn.
- To introduce unsupervised learning techniques such as K-Means Clustering and their applications.
- To enable students to analyze, evaluate, and compare the performance of different machine learning algorithms through mini projects.

Course Outcomes: After completion of the course, students should be able to

- **I**mplement statistical measures and apply Python libraries for data analysis.
- Utilize data processing and visualization tools such as Pandas and Matplotlib effectively.
- Apply supervised machine learning algorithms including Regression, Decision Trees, and KNN using sklearn.
- Demonstrate the use of unsupervised learning techniques such as K-Means clustering.
- Analyze the performance of machine learning models on real-world datasets through mini projects.

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.
11. Implement decision tree algorithm.
12. Implement Support Vector Machine for the sample data.
13. Write a python program to implement Hidden Markov models.
14. Implementation of Linear Discriminant Analysis.

TEXT BOOK:

1. Machine Learning – Tom M. Mitchell, - MGH.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis.

III YEAR II SEMESTER (VI SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2460518	Data Mining	PC	3	0	0	3	40	60	100
2	2466601	Soft Computing	PC	3	1	0	4	40	60	100
3	2460517	Compiler Design	PC	3	1	0	4	40	60	100
4		Professional Elective-II	PE	3	0	0	3	40	60	100
5		Open Elective-II	OE	3	0	0	3	40	60	100
		Laboratory								
1	2466675	Augmented Reality and virtual Reality laboratory	PC	0	0	2	1	40	60	100
2	2466671	Soft Computing Laboratory	PC	0	0	2	1	40	60	100
3	2460586	Data Mining Laboratory	PC	0	0	2	1	40	60	100
		Mandatory Course								
1	2460023	Constitution of India	MC	0	0	0	0	-	-	-
Total Credits				**15**	**2**	**6**	**20**	**320**	**480**	**800**

2460518: DATA MINING

L	T	P	C
3	0	0	3

Prerequisites: Database Management System, Mathematical statistical Foundation

Course Objectives:

- To introduce the fundamental concepts of data mining, knowledge discovery, and data preprocessing techniques.
- To familiarize students with association analysis and pattern mining algorithms such as Apriori and FP-Growth.
- To develop the ability to implement and evaluate classification techniques including Decision Trees, Bayes Classifiers, and SVMs.
- To provide knowledge of clustering methods, outlier detection, and their applications in analyzing complex datasets.
- To expose students to advanced data mining concepts such as Web Mining, Spatial Mining, and Temporal Mining for real-world applications.

Course Outcomes: After completion of the course, students should be able to

- Understand fundamental concepts of data mining, data preprocessing, and similarity measures for real-world datasets.
- Implement association analysis techniques like Apriori and FP-Growth for discovering meaningful patterns in data.
- Apply classification algorithms including Decision Trees, Bayes Methods, SVM, and Neural Networks to solve predictive problems.
- Demonstrate clustering techniques, outlier detection, and evaluation methods for grouping and analyzing complex data.
- Analyze advanced data mining concepts such as Web, Spatial, and Temporal Mining for domain-specific applications.

Module- I **[10]**

Introduction to Data Mining: What Data mining? Kinds of Data, Knowledge Discovery process, Data Mining Functionalities, Kinds of Patterns, Major Issues in Data Mining. Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity, Data Pre-processing:

Major Tasks in Data Pre-processing, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

Module- II **[10]**

Association **Analysis:** Basic Concepts, Market Basket Analysis, Apriori Algorithm, FP-growth, From Association Analysis to Correlation Analysis, Pattern Mining in Multilevel Associations and Multidimensional Associations.

Module- III **[10]**

Classification: Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Metrics for Evaluating Classifier Performance, Ensemble Methods, Multilayer Feed- Forward Neural Network, Support Vector Machines, k-Nearest-Neighbor Classifiers.

Module- IV **[10]**

Cluster Analysis: Requirements for Cluster Analysis, Overview of Basic Clustering Methods, Partitioning Methods-k-Means, k-Medoids, Hierarchical Methods-AGENES, DIANA, BIRCH, Density- Based Method-DBSCAN, Outlier Analysis: Types of Outliers, Challenges of Outlier Detection, and Overview of Outlier Detection Methods

Module- V **[8]**

Advanced Concepts: Web Mining- Web Content Mining, Web Structure Mining, Web Usage Mining, Spatial Mining- Spatial Data Overview, Spatial Data Mining Primitives, Spatial Rules, Spatial Classification Algorithm, Spatial Clustering Algorithms, Temporal Mining- Modeling Temporal Events, Time Series, Pattern Detection, Sequences, Temporal Association Rules.

Text Books:

1. Jiawei Han, Micheline Kamber, Jian Pei., Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann/Elsevier, 2012.
2. Margaret H Dunham, Data Mining Introductory and Advanced Topics, 2nd Edition, Pearson Education, India, 2006.

Reference Books:

1. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities

Press.
2. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne and Vipin Kumar, Introduction to Data Mining, 2nd Edition, Pearson Education India, 2021.
3. Amitesh Sinha, Data Warehousing, Thomson Learning, India, 2007.

2466601: SOFT COMPUTING

L	T	P	C
3	1	0	4

Prerequisites: Machine Learning

Course Objectives:

- To familiarize with soft computing concepts
- To introduce and use the idea of fuzzy logic and use of heuristics based on human experience
- To familiarize the Neuro-Fuzzy modeling using Classification and Clustering techniques
- To learn the concepts of Genetic algorithm and its applications
- To acquire the knowledge of Rough Sets.

Course Outcomes: After completion of the course, students should be able to

- Explain the principles, evolution, and characteristics of soft computing and differentiate it from hard computing.
- Analyze uncertain and imprecise data using fuzzy sets, relations, and rule-based systems.
- Implement fuzzy decision-making techniques and particle swarm optimization for real-world problem solving.
- Develop solutions using genetic algorithms, including encoding, selection, crossover, mutation, and fitness evaluation.
- Apply rough sets, rule induction, and combine multiple soft computing methods for complex problem-solving.

Module – I **[10]**

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.

Module – II **[10]**

Fuzzy Systems: Fuzzy Sets, Fuzzy Relations, Fuzzy Logic, Fuzzy Rule-Based Systems, Fuzzy Decision Making, Fuzzification & Defuzzification Methods

Module – III **[8]**

Fuzzy Inference Systems (FIS),Mamdani vs. Sugeno FIS, Rule-Based Fuzzy Systems, Applications: Autonomous Vehicles, Washing Machines, Decision Support Systems

Module – IV **[10]**

Genetic Algorithms: Basic Concepts, Basic Operators for Genetic Algorithms, Crossover and Mutation Properties, Genetic Algorithm Cycle, Fitness Function, Applications of Genetic Algorithm.

Module – V **[10]**

Rough Sets, Rough Sets, Rule Induction, and Discernibility Matrix, Integration of Soft Computing Techniques.

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

2460517: COMPILER DESIGN

L	T	P	C
3	1	0	4

Prerequisites: Formal Languages and Automata Theory

Course Objectives:

- To understand the structure, phases, and role of a compiler in translating programming languages.
- To learn lexical analysis techniques using regular expressions, finite automata, and lexical analyzer generators.
- To study syntax analysis methods, grammar construction, and parsing techniques for language processing.
- To explore syntax-directed translation, intermediate code generation, type checking, and run-time environments.
- To analyze and apply code generation strategies and machine-independent optimization techniques for efficient execution.

Course Outcomes: After completion of the course, students should be able to

- Implement lexical analyzers using regular expressions, finite automata, and Lex tools.
- Apply parsing techniques (top-down, bottom-up, LR) for analyzing context-free grammars.
- Construct syntax-directed translations and generate intermediate code for programming constructs.
- Design run-time environments and apply code generation strategies with optimization.
- Perform machine-independent code optimization using data-flow analysis and loop optimizations.

Module-I **[10]**

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.

Module- II **[10]**

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.

Module– III **[10]**

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.

Module- IV **[10]**

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.

Module- V **[8]**

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.

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman.

References:

1. Lex&Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction by Louden, Thomson

24X6645: NETWORK SECURITY

L T P C
3 0 0 3

Course Objectives:

- To introduce fundamental concepts of network security and cryptography, including security services, mechanisms, and models.
- To develop an understanding of symmetric and asymmetric encryption algorithms, hash functions, and key distribution techniques.
- To apply cryptographic principles for securing data in real-time communication, transport-level protocols, wireless, and email security.
- To analyze different security attacks, vulnerabilities, and corresponding countermeasures in network and information systems.
- To enable students to evaluate and implement cryptographic techniques through case studies on secure transactions and digital security applications.

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

- Understand basic cryptographic algorithms, message and web authentication and security issues.
- Identify information system requirements for both client and server.
- Determine client-side and server-side requirements for information system design.
- Implement encryption, decryption, hashing, and authentication protocols using suitable cryptographic algorithms.
- Compare the effectiveness of transport-level, wireless, and email security mechanisms against different attack scenarios.

MODULE – I [10]

Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms

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.

MODULE – II [10]

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, Diffie-Hellman Key Exchange, Knapsack Algorithm.

MODULE – III [10]

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

MODULE – IV [10]

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

MODULE – V [8]

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

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

24X6646: INFORMATION RETRIEVAL SYSTEMS

L T P C
3 0 0 3

Prerequisites: 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.
- Develop knowledge of automatic indexing methods and clustering techniques for effective organization and retrieval of information.
- Explore user search strategies, similarity measures, ranking methods, and visualization tools to enhance information access and usability.
- Provide an understanding of text search algorithms and multimedia information retrieval techniques, covering text, audio, image, and video data.

Course Outcomes: After completion of the course, student should be able to

- Explain the objectives, functional capabilities, and scope of Information Retrieval Systems and their integration with digital libraries and databases.
- Analyze different document clustering algorithms
- Implement retrieval systems for web search tasks.
- Design Information Retrieval System for web search tasks.
- Develop retrieval solutions for multimedia data such as speech, audio, images, and video.

MODULE – I [10]

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

MODULE – II [10]

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.

MODULE – III [10]

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

MODULE – IV [10]

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

MODULE – V [8]

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

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.

24X6644: Digital Image Processing

L T P C
3 0 0 3

Course Objectives

- To explore image acquisition, sampling and quantization, preprocessing, enhancement, restoration, segmentation and compression.
- To understand fundamentals of digital images, imaging geometry, and 2D transformations such as DFT, DCT, KLT, and SVD.
- To explore spatial and frequency domain techniques for image enhancement and filtering.
- To apply restoration techniques including inverse filtering, least mean square filters, and constrained restoration.
- To analyze image segmentation methods, including edge detection, thresholding, and region-based approaches.

Course Outcomes: After completion of the course, student should be able to

- Demonstrate the basic concepts of two-dimensional signal acquisition, sampling, and quantization.
- Understand the knowledge of filtering techniques.
- Implement the knowledge of image enhancement, segmentation, restoration and compression techniques.
- Analyze segmentation methods including edge detection, boundary linking, and thresholding techniques.
- Evaluate image compression models, distinguishing between error-free and lossy compression, and proposing efficient coding strategies.

MODULE – I [10]

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.

MODULE – II [10]

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

MODULE – III [10]

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

MODULE – IV [10]

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

MODULE – V [8]

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

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

24X0527: IOT COMMUNICATION PROTOCOLS

L	T	P	C
3	0	0	3

Course Objectives:

- In this course, learners will be going to learn about various protocols designed for the implementation of the Internet of Things (IoT) applications.
- To understand fundamentals of IoT, including architecture, standards, devices, gateways, networking, and analytics.
- To explore IoT reference architectures, functional, information, and deployment views, along with design constraints.
- To apply knowledge of IoT Data Link Layer protocols such as IEEE 802.11, Zigbee, BLE, DASH7, and WirelessHART in practical scenarios.
- To analyze Network Layer protocols, including IPv4/IPv6, 6LoWPAN, RPL, CARP, and related addressing and routing methods.

Course Outcomes: After Completion of the course, student should be able to

- Understand the IoT architecture, technology fundamentals, devices, gateways, and business process concepts.
- Explain various architectural views, including functional, information, and operational perspectives.
- Implement Data Link Layer protocols in IoT networks using practical examples.
- Analyze Network Layer protocols, comparing their performance, addressing, and routing strategies.
- Evaluate Transport and Session Layer protocols, proposing secure and optimized solutions for IoT communication.

MODULE – I [10]

Introduction: IoT architecture outline, standards - IoT Technology Fundamentals- Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service (XaaS), M2M and IoT Analytics

MODULE – II [10]

IoT Reference Architecture: Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant Architectural

views. Real-World Design Constraints- Introduction, Technical Design constraints

MODULE – III [10]

IoT Data Link Layer: PHY/MAC Layer (3GPP MTC, IEEE 802.11, IEEE 802.15), Wireless HART, ZWave, Bluetooth Low Energy, Zigbee Smart Energy, DASH7

MODULE – IV [10]

Network Layer Protocols: Network Layer-IPv4, IPv6, 6LoWPAN, 6TiSCH, ND, DHCP, ICMP, RPL, CORPL, CARP

MODULE – V [8]

IoT Transport & Session Layer Protocols: Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP)- (TLS, DTLS) – Session Layer-HTTP, CoAP, XMPP, AMQP, MQTT

TEXT BOOKS:

1. Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications ,2016

2. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence",1st Edition, Academic Press, 2015

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.

2466675: AUGMENTED REALITY AND VIRTUAL REALITY LABORATARY

L	T	P	C
0	0	2	1

COURSE OUTCOMES: After Completion of the course, student should be able to

- Understand Unity environment and interface for creating 2D/3D applications.
- Develop interactive games with physics and animation using Unity.
- Implement UI elements and score systems in game environments.
- Explore AR/VR platforms and develop basic immersive applications.
- Design complete game applications using Unity.

LIST OF EXPERIMENTS

1. Install Unity and Visual Studio. Explore Unity 2D/3D templates. Overview of Unity Editor: Scene, Game, Hierarchy, Project, Inspector, Game Objects, Components.

2. Create simple 3D objects. Apply transformations. Add basic lighting, materials, and shaders. Understand how to use prefabs and assets.

3. Add Rigidbody, Colliders, Physics Materials. Apply forces and detect collisions using OnCollisionEnter. Demonstrate physics joints and triggers.

4. Set up project structure with folders (Materials, Prefabs, Scripts, Scenes). Create environment and moving player. Write movement scripts.

5. Move the camera, build play area boundaries. Create and collect pick-up objects (gems), update score, and display on screen using UI Text.

6. Add an enemy to follow the player using NavMesh or scripting. Create a health bar and display it. Show “Game Over” and “You Win” conditions.

7. Create a complete UI with canvas, buttons, health bar, score, and pop-ups. Script UI interactions using C#.

8. Build the Infinite Runner game into a Windows executable. Package

resources. Test for debugging and playability.

9. Define AR & VR with real-time examples. Introduction to tools like Oculus, Vuforia, Kudan, Wikitude, ARKit, and ARCore.

10 Explore VR environment using Oculus. Experience basic interactions and movement in virtual environments.

11 Install Vuforia in Unity, generate license key, create and configure database. database. Place 3D models on image targets.

12 Interact with augmented objects in the real world. Build and test AR application on mobile device or simulator.

TEXTBOOKS

1. Steven M. LaValle, Virtual Reality, Cambridge University Press, 2016.
2. William R. Sherman & Alan B. Craig, Understanding Virtual Reality, Morgan Kaufmann, 2002.
3. Alan B. Craig et al., Developing Virtual Reality Applications, Morgan Kaufmann, 2009.
4. Allan Fowler, AR Game Development, Apress, 2018. ISBN: 978-1484236178.
5. Schmalstieg & Hollerer, Augmented Reality: Principles & Practice, Pearson, 2016. ISBN-10: 933257849.

REFERENCE BOOKS

1. Gerard Jounghyun Kim, Designing Virtual Systems, 2005.
2. Doug A. Bowman et al., 3D User Interfaces: Theory and Practice, Addison Wesley, 2005.
3. Oliver Bimber & Ramesh Raskar, Spatial Augmented Reality, 2005.
4. Grigore C. Burdea & Philippe Coiffet, Virtual Reality Technology, Wiley, 2003.
5. Kharis O'Connell, Designing for Mixed Reality, O'Reilly Media, 2016.
6. Sanni Siltanen, Theory and Applications of Marker-Based AR, 2012. ISBN: 978-951-38-7449-0.

2466671: SOFT COMPUTING LAB

L	T	P	C
0	0	2	1

Prerequisites: Machine learning lab

Course Objectives:

- Apply basic learning algorithms like Perceptron, Hebb's rule, and Delta rule.
- Design and implement ANNs with and without backpropagation.
- Develop models using regression, logic gates, and SVM with fuzzy concepts.
- Perform operations on fuzzy sets and relations.
- Solve optimization problems using genetic algorithms.

Course Outcomes: After completion of the course, students should be able to

- Implement perceptron, Hebb's, and Delta learning rules.
- Design neural networks with and without backpropagation.
- Apply regression and logic gate models.
- Perform fuzzy set and relation operations.
- Use SVM with fuzzy concepts and genetic algorithms.

List of Experiments:

1. Implement linear regression and Multi-Linear regression.
2. Implement Logistic regression.
3. Implement Union, Intersection, Complement and Difference operations on fuzzy sets.
4. create fuzzy relation by Cartesian product of any two fuzzy sets and perform max-min composition on any two fuzzy relations.
5. Implement at least three defuzzification methods — Centroid (COG), Mean of Maxima (MOM), and Bisector — and demonstrate differences on the same fuzzy output set
6. Given a CSV of a real-valued feature (e.g., temperature readings), implement an automated procedure to fit triangular membership functions for linguistic labels (low/med/high) using k-means or percentiles.
7. Implement the same control problem (e.g., cruise-control throttle) in both Mamdani and Sugeno frameworks
8. Implement SVM classification by Fuzzy concepts.
9. Build a fuzzy controller for a washing machine that decides wash time and spin speed from: dirt level, load size, and fabric type. Provide rule base, membership functions, GUI/console demo, and simulated cycles.

10. Create a small decision support prototype for loan approval using fuzzy rules combining credit-score, income stability, and existing obligations. Produce an explainable rule-activation log for each decision.
11. Implement a GA to maximize/minimize a multimodal benchmark function (e.g., Rastrigin or Rosen brock). Include binary and real-valued encodings, selection (tournament), crossover (single/differential), mutation, and elitism.
12. Write a small interactive script to visualize crossover (one-point, two-point, uniform) and mutation effects on sample chromosomes. Show offspring generation step-by-step and compute Hamming distances.
13. Implement travelling sales person problem (TSP) using genetic algorithms.
14. Implement the discernibility matrix for a small tabular dataset (discrete attributes). From it, derive reduces and induce decision rules. Validate induced rules on test instances and report coverage/accuracy.

Textbooks:

1. Principles of Soft Computing By S.N. Sivanandam, S.N.Deepa Wiley 3rd Edition

Reference books:

1. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic - Theory and Applications, Prentice Hall, 1995.
2. Ross J.T., Fuzzy Logic with Engineering Applications John Wiley & Sons, 2009

2460586: DATA MINING LABORATORY

L	T	P	C
0	0	2	1

Prerequisites: Database Management Systems Laboratory

Course Objectives:

- To understand and apply various data preprocessing techniques including cleaning, transformation, and integration to prepare datasets for analysis.
- To explore data partitioning and warehouse schemas to efficiently organize and manage large-scale data.
- To implement data mining algorithms such as association rule mining, decision trees, and clustering methods to extract meaningful patterns.
- To perform classification and prediction using techniques like Bayesian classification and K-nearest neighbor methods.
- To develop skills in OLAP operations and data cube construction for multidimensional data analysis and decision-making.

Course Outcomes: After completion of the course, students should be able to

- **A**pply data preprocessing techniques to clean, transform, and integrate datasets for effective analysis.
- Implement data warehousing solutions using different schemas and data partitioning methods.
- Execute data mining algorithms like Apriori, FP-Growth, and decision trees to discover patterns and relationships in data.
- Perform data classification and clustering using methods such as Bayesian, KNN, K-Means, BIRCH, PAM, and DBSCAN.
- Perform OLAP operations to analyze multidimensional data for informed decision-making.

LIST OF EXPERIMENTS: Experiments using Weka/ Python

1. Data Processing Techniques:

   (i) Data cleaning (ii) Data transformation – Normalization (iii) Data integration

2. Partitioning - Horizontal, Vertical, Round Robin, Hash based
3. Data Warehouse schemas – star, snowflake, fact constellation
4. Data cube construction – OLAP operations
5. Data Extraction, Transformations & Loading operations
6. Implementation of Attribute oriented induction algorithm
7. Implementation of apriori algorithm
8. Implementation of FP – Growth algorithm
9. Implementation of Decision Tree Induction
10. Calculating Information gain measures
11. Classification of data using Bayesian approach
12. Classification of data using K – nearest neighbour approach
13. Implementation of K – means algorithm
14. Implementation of BIRCH algorithm
15. Implementation of PAM algorithm
16. Implementation of DBSCAN algorithm

Text Books:

1. Data Mining – Concepts and Techniques - JIAWEI HAN & MICHELINE KAMBER, Elsevier.
2. Data Warehousing, Data Mining &OLAP- Alex Berson and Stephen J. Smith- Tata McGraw-Hill Edition, Tenth reprint 2007

Reference Book:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Anuj Karpatne, Introduction to Data Mining, Pearson Education

2460023: CONSTITUTION OF INDIA

B.Tech. II Year I Sem.

L/T/P/D/C

3/0/0/0/0

The Constitution of India is the supreme law of India. Parliament of India cannot make any law which violates the Fundamental Rights enumerated under the Part III of the Constitution. The Parliament of India has been empowered to amend the Constitution under Article 368, however, it cannot use this power to change the "basic structure" of the constitution, which has been ruled and explained by the Supreme Court of India in its historical judgments. The Constitution of India reflects the idea of "Constitutionalism" – a modern and progressive concept historically developed by the thinkers of "liberalism" – an ideology which has been recognized as one of the most popular political ideology and result of historical struggles against arbitrary use of sovereign power by state. The historic revolutions in France, England, America and particularly European Renaissance and Reformation movement have resulted into progressive legal reforms in the form of "constitutionalism" in many countries. The Constitution of India was made by borrowing models and principles from many countries including United Kingdom and America.

The Constitution of India is not only a legal document but it also reflects social, political and economic perspectives of the Indian Society. It reflects India's legacy of "diversity". It has been said that Indian constitution reflects ideals of its freedom movement; however, few critics have argued that it does not truly incorporate our own ancient legal heritage and cultural values. No law can be "static" and therefore the Constitution of India has also been amended more than one hundred times. These amendments reflect political, social and economic developments since the year 1950. The Indian judiciary and particularly the Supreme Court of India has played an historic role as the guardian of people. It has been protecting not only basic ideals of the Constitution but also strengthened the same through progressive interpretations of the text of the Constitution. The judicial activism of the Supreme Court of India and its historic contributions has been recognized throughout the world and it gradually made it "as one of the strongest court in the world".

Course content

1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution of India
3. Salient features and characteristics of the Constitution of India

4. Scheme of the fundamental rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation
7. Federal structure and distribution of legislative and financial powers between the Union and theStates
8. Parliamentary Form of Government in India – The constitution powers and status of thePresident of India
9. Amendment of the Constitutional Powers and Procedure
10. The historical perspectives of the constitutional amendments in India
11. Emergency Provisions: National Emergency, President Rule, Financial Emergency
12. Local Self Government – Constitutional Scheme in India
13. Scheme of the Fundamental Right to Equality
14. Scheme of the Fundamental Right to certain Freedom under Article 19
15. Scope of the Right to Life and Personal Liberty under Article 21

IV YEAR I SEMESTER (VII SEMESTER)

S.No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
		Theory								
1	2470541	Deep Learning	PC	2	0	0	2	40	60	100
2	2470544	Full Stack Development	PC	3	0	0	3	40	60	100
3	2470010	Business Economics and Financial Analysis	HSMC	3	0	0	3	40	60	100
4		Professional Elective-III	PE	3	0	0	3	40	60	100
5		Professional Elective-IV	PE	3	0	0	3	40	60	100
6		Open Elective-III	OE	3	0	0	3	40	60	100
		Laboratory								
1	2476672	Deep Learning Laboratory	PC	0	0	2	1	40	60	100
2	2470580	Full Stack Development Laboratory	PC	0	0	2	1	40	60	100
	2470588	**Internship**	PS	0	0	2	1	100	-	100
		Project								
1	2470590	Project Stage –I	PS	0	0	6	3	100	-	100
		Mandatory								
	2470025	Human Values and Professional Ethics	MC	0	0	0	0	-	-	-
Total Credits				**17**	**0**	**12**	**23**	**520**	**480**	**1000**

2470541: DEEP LEARNING

L	T	P	C
2	0	0	2

Prerequisites: Machine Learning

Course Objectives:

- To understand basics of machine learning and deep feedforward networks.
- To learn regularization and optimization methods for deep models.
- To study convolutional networks and their applications.
- To explore recurrent and recursive networks for sequence data.
- To apply deep learning techniques to real-world problems in vision, speech, and NLP.

Course Outcomes: After completion of the course, students should be able to

- Implement and train deep feedforward neural networks using gradient-based learning.
- Apply regularization and optimization techniques to improve model performance.
- Design and implement convolutional neural networks for image and structured data tasks.
- Develop recurrent and recursive neural networks for sequence modeling and long-term dependencies.
- Evaluate and apply deep learning models to real-world applications in computer vision, speech, and NLP.

Module – I **[10]**

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

Module – II **[10]**

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

Module – III **[10]**

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

Module – IV **[10]**

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

Module – V **[8]**

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.

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.

24X0544: FULL STACK DEVELOPMENT

L T P C
3 0 0 3

Pre-Requisites: Object Oriented Programming using JAVA

Course Outcomes: After completion of the course, students should be able to

- Understand Full stack components for developing web application.
- Apply packages of NodeJS to work with Data, Files, HttpRequests and Responses.
- Use MongoDB data base for storing and processing huge data and connects with Node JS application.
- Design faster and effective single page applications using Express and Angular.
- Create interactive user interfaces with react components.

Module-I **[10]**

Introduction to Full Stack Development:

Understanding the Basic Web Development Framework- User, Browser, Web server, Backend Services, Full Stack Components - Node.js, MongoDB, Express, React, Angular. Java Script Fundamentals, Node JS- 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

Module-II **[10]**

Node.js:

Working with JSON, Using the BufferModule to BufferData, 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.

Module-III **[10]**

MongoDB:

NeedofNoSQL,UnderstandingMongoDB,MongoDBDataTypes,PlanningYourDataModel,Building the Mongo DB 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

Module-IV **[10]**

Express and Angular:

Getting Started with Express, Configuring Routes, Using Requests Objects, Using ResponseObjects. Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components,Expressions,DataBinding,Built-inDirectives,CustomDirectives,ImplementingAngular Services in Web Applications.

Module-V **[8]**

React:

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components,CreatingComponentsinReact,DataandDataFlowinReact, RenderingandLifeCycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

TEXTBOOKS:

1. BradDayley, Brendan Dayley, Caleb Dayley.,Node.js,MongoDBandAngularWeb Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas,Reactin Action,1st Edition, Manning Publications.

REFERENCEBOOKS:

1. VasanSubramanian,ProMERNStack,FullStackWebAppDevelopmentwithMongo, Express, React, and Node, 2ndEdition, Apress, 2019.
2. ChrisNorthwood,TheFullStackDeveloper:YourEssentialGuidetotheEverydaySkills Expected of a Modern Full Stack Web Developer', 1stedition, Apress, 2018.
3. KirupaChinnathambi,LearningReact:AHands-

OnGuidetoBuildingWebApplicationsUsing React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

24X0010: BUSINESS ECONOMICS AND FINANCIAL ANALYSIS

B.Tech. IV Year I 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 in 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 & Law of Supply.

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

Production Analysis: Factors of Production, Production Function, Production Function with onevariable 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 ProfitAnalysis.

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.

UNIT – V: Financial Analysis through Ratios: Concept of Ratio Analysis, Liquidity Ratios, TurnoverRatios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios (simple problems).Introduction to Fund Flow and Cash Flow Analysis (simple problems).

TEXT BOOKS:

1. D. D. Chaturvedi, S. L. Gupta, Business Economics - Theory and Applications, InternationalBook House Pvt. Ltd. 2013.
2. Dhanesh K Khatri, Financial Accounting, Tata McGraw Hill, 2011.
3. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, Managerial Economics, 2e, Tata McGrawHill 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, VikasPublications, 2013.

24X0528: WEB SECURITY

L T P C
3 0 0 3

Course Objectives:

- To give an Overview of information security
- Introduce the fundamentals of web security, risk analysis, cryptographic systems, protocols, and digital identification methods.
- Understand the concepts of access control for relational databases.
- Develop an understanding of privacy protection, server security, and web application security, along with best practices for safeguarding systems.
- Explore database security mechanisms, including access control models, trust management, and securing data warehouses/OLAP systems.

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

- Understand how common mistakes can be bypassed and exploit the application
- Design the client side and service side programming
- Identify common application vulnerabilities
- Apply database access control models and assess trust management approaches for securing data warehouses and OLAP systems.
- Investigate future trends in database and web security, and propose privacy-preserving solutions for mobile and location-based environments.

MODULE – I [10]

The Web Security, The Web Security Problem, Risk Analysis and Best Practices Cryptography and the Web: Cryptography and Web Security, Working Cryptographic Systems and Protocols, Legal Restrictions on Cryptography, Digital Identification

MODULE – II [10]

The Web's War on Your Privacy, Privacy-Protecting Techniques, Backups and Antitheft, Web Server Security, Physical Security for Servers, Host Security for Servers, Securing Web Applications

MODULE – III [10]

Database Security: Recent Advances in Access Control, Access Control Models for XML, Database Issues in Trust Management and Trust Negotiation, Security in Data Warehouses and OLAP Systems

MODULE – IV [10]

Security Re-engineering for Databases: Concepts and Techniques, Database Watermarking for Copyright Protection, Trustworthy Records Retention, Damage Quarantine and Recovery in Data Processing Systems, Hippocratic Databases: Current Capabilities.

MODULE – V [8]

Future Trends Privacy in Database Publishing: A Bayesian Perspective, Privacy-enhanced Location Based Access Control, Efficiently Enforcing the Security and Privacy Policies in a Mobile Environment

TEXT BOOKS:

1. Web Security, Privacy and Commerce Simson G Arfinkel, Gene Spafford, O'Reilly.

2. Handbook on Database security applications and trends Michael Gertz, Sushil Jajodia

24X0533: DATA ANALYTICS

L T P C
3 0 0 3

Prerequisites: Database management Systems

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.

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

- Understand the impact of data analytics for business decisions and strategy
- Perform data analysis/statistical analysis
- Do standard data visualization and formal inference procedures
- Design Data Architecture
- Utilize various Data Sources

Module – I **[10]**

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.

Module – II **[10]**

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. Need for Business Modeling.

Module – III **[10]**

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.

Module – IV **[8]**

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

Module – V **[10]**

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

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, Addison 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 LabsJeffrey D Ullman Stanford Univ.

24X6603: EXPERT SYSTEMS

L T P C
3 0 0 3

Course Objectives:

- Understand the basic techniques of artificial intelligence.
- Knows the Non-monotonic reasoning and statistical reasoning.
- Provide a thorough understanding of fundamental artificial intelligence techniques including search strategies, game playing, and heuristic algorithms.
- Introduce knowledge representation methods such as predicate logic, semantic nets, frames, rules-based systems, and constraint propagation.
- Explain the concepts, architecture, and organization of expert systems and their application to problem-solving.

Course Outcomes: After completion of the course, student should be able to

- Apply the basic techniques of artificial intelligence.
- Discuss the architecture of an expert system and its tools.
- Understand the importance of building an expert systems
- Apply basic AI techniques, including blind search, heuristic search, and game-playing algorithms.
- Demonstrate various knowledge representation techniques and implement rules-based deduction systems for structured problem-solving.

MODULE – I [10]

Introduction to AI programming languages, Blind search strategies, Breadth-first – Depth-first –Heuristic search techniques Hill Climbing – Best first – A Algorithms AO* algorithm – game trees, Minmax algorithms, game playing – Alpha-beta pruning.

MODULE – II [10]

Knowledge representation issues predicate logic – logic programming Semantic nets- frames and inheritance, constraint propagation; Representing Knowledge using rules, Rules-based deduction systems.

MODULE – III [10]

Introduction to Expert Systems, Architecture of expert systems, Representation and organization of knowledge, Basics characteristics, and types of problems handled by expert systems.

MODULE – IV [10]

Expert System Tools: Techniques of knowledge representations in expert systems, knowledge engineering, system-building aids, support facilities, stages in the development of expert systems.

MODULE – V [8]

Building an Expert System: Expert system development, Selection of the tool, Acquiring Knowledge, Building process.

Problems with Expert Systems: Difficulties, common pitfalls in planning, dealing with domain experts, difficulties during development.

TEXT BOOKS:

1. Elain Rich and Kevin Knight, “Artificial Intelligence”, Tata McGraw-Hill, New Delhi.

2. Waterman D.A., “A Guide to Expert Systems”, Addison Wesley Longman.

REFERENCE BOOKS:

1. Stuart Russel and other Peter Norvig, “Artificial Intelligence – A Modern Approach”, PrenticeHall,

2. Patrick Henry Winston, “Artificial Intelligence”, Addison Wesley,

3. Patterson, Artificial Intelligence & Expert System, Prentice Hall India, 1999.

4. Hayes-Roth, Lenat, and Waterman: Building Expert Systems, Addison Wesley,

5. Weiss S.M. and Kulikowski C.A., “A Practical Guide to Designing Expert Systems”, Rowman & Allanheld, New Jersey.

24X0531: **IOT SYSTEM ARCHITECTURES**

L T P C
3 0 0 3

Prerequisites: Internet of Things

Course Outcomes: After Completion of the course, student should be able to

- Understand IoT applications and IoT Architectures.
- Learn about IoT devices and event driven analysis
- Analyze Industrial IOT.
- Apply Industrial IOT for safety and security
- Perform security testing of IoT systems

Module - I: **[8]**

The IoT Landscape: What Is IoT? Applications, Architectures, Wireless Networks, Devices, Security and Privacy, Event-Driven Systems. **IoT System Architectures:** Introduction, Protocols Concepts, IoT- Oriented Protocols, Databases, Time Bases, Security.

Module - II: **[10]**

IoT Devices & Event-Driven System Analysis: The IoT Device Design Space, Cost of Ownership and Power Consumption, Cost per Transistor and Chip Size, Duty Cycle and Power Consumption, Platform Design. **Event-Driven System Analysis:** Introduction, Motivating Example, IoT Network Model, Events, Networks, Devices and Hubs, Single-Hub Networks, Multi-hub Networks, Network Models and Physical Networks, IoT Event Analysis, Event Populations, Stochastic Event Populations, Environmental Interaction Modeling, Event Transport and Migration.

Module – III **[10]**

Industrial Internet of Things: Introduction, Industry 4.0, Industrial Internet of Things (IIOT), IIOT Architecture, Basic Technologies, Applications and Challenges.

Module - IV: **[10]**

Security and Safety: Introduction, Systems Security, Network Security, Generic Application Security, Application Process Security and Safety, Reliable-and-Secure-by-Design IoT Applications, Run-Time Monitoring, The ARMET Approach, Privacy and Dependability.

Module - V: **[10]**

Security Testing IoT Systems: Introduction, Fuzz Testing for Security, White-Box Fuzzing, Black-Box Fuzzing, Fuzzing Industrial Control Network Systems, Fuzzing Modbus, The Modbus Protocol, Modbus/TCP Fuzzer.

TEXT BOOKS:

1. Dimitrios Serpanos, Marilyn Wol, Internet-of-Things (IoT) Systems Architectures, Algorithms, Methodologies, ISBN 978-3-319-69714-7.

REFERENCE BOOKS:

1. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015.
2. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi and Wiley, 2012 (for Unit 2).
3. "From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence", Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle and Elsevier, 2014.
4. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017.
5. "Building the Internet of Things" by Maciej Kranz

24X0540: BLOCKCHAIN TECHNOLOGY

L T P C
3 0 0 3

Prerequisites: cryptography and Network security, 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.
- Exploring cryptocurrency principles, Bitcoin, altcoins, tokens, and their usage in digital economy.
- Applying knowledge of public, private, and consortium blockchains for real-world problem-solving.

Course Outcomes: After completion of the course, students should be able to

- Understand concepts behind crypto currency
- Apply of smart contracts in decentralized application development
- Explain public, private, and consortium blockchain architectures with relevant use cases.
- Implement smart contracts and blockchain applications using Ethereum, Python, or Hyperledger Fabric.
- Analyze blockchain security, privacy, scalability, and performance challenges across domains.

MODULE – I [10]

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.

MODULE – II [10]

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.

MODULE – III [10]

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.

MODULE – IV [10]

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.

MODULE – V [8]

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.

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.

24X0532: CLOUD COMPUTING

L T P C
3 0 0 3

Pre-requisites: Computer Networks, Operating System.

Course Objectives:

- To understand computing paradigms, cloud fundamentals, architecture, and management principles.
- To explore cloud deployment models, service models, and the technological drivers supporting cloud computing.
- To apply virtualization techniques and programming models such as MapReduce and Cloud Haskell for cloud-based software development.
- To analyze networking aspects of cloud computing including data center environments, transport layer issues, and cloud service providers.
- To evaluate security concerns and advanced concepts in cloud computing for research and industrial applications.

Course Outcomes: After completion of course, students should be able to

- Understand different computing paradigms and potential of the paradigms and specifically cloud computing
- Describe cloud service types, cloud deployment models and technologies supporting and driving the cloud
- Implement the knowledge of programming models for cloud and development of software application that runs the cloud and various services available from major cloud providers
- Analyze the security concerns and issues in cloud computing
- Acquire the knowledge of advances in cloud computing.

MODULE – I [10]

Computing Paradigms, Cloud Computing Fundamentals, Cloud Computing Architecture and Management

MODULE – II [10]

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

MODULE – III [10]

Virtualization, Programming Models for Cloud Computing: MapReduce, Cloud Haskell, Software Development in Cloud

MODULE – IV [10]

Networking for Cloud Computing: Introduction, Overview of Data Center Environment, Networking Issues in Data Centers, Transport Layer Issues in DCNs, Cloud Service Providers

MODULE – V [8]

Security in Cloud Computing, and Advanced Concepts in Cloud Computing

TEXT BOOK:

1.Chandra sekaran, 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

24X0538: COMPUTER VISION &ROBOTICS

L T P C
3 0 0 3

Course outcomes: After successful completion of the course, students should be able to

- Implement fundamental image processing techniques used in computer vision.
- Use boundary tracking and region descriptors.
- Apply codes for Hough Transform (line, circle, ellipse).
- Analyze 3D vision techniques and implement motion-related methods.
- Develop computer vision applications.

Module-I: **[10]**

CAMERAS:-Pinhole Cameras

Radiometry :- Measuring Light in Space, Light Surfaces, Important Special Cases, Sources, Shadows, And Shading:- Qualitative Radiometry, Irradiance, The Irradiance Equation, Ideal Shading Models, Applications: Photometric Stereo, Interreflections: Global Shading Models

Color: The Physics of Color, Human Color Perception, Representing Color, A Model for Image Color, Surface Color from Image

Module-II **[10]**

Linear Filters : Linear Filters and Convolution, Shift Invariant Linear Systems, Spatial Frequency and Fourier Transform

Edge Detection: Noise, Estimating Derivatives, Detecting Edges
Texture: Representing Texture, Analysis and Synthesis Using Oriented Pyramids, Applications: Synthesizing Textures, Shape from Texture

Module-III: **[8]**

The Geometry of Multiple Views: Two Views

Stereopsis: Binocular Stereopsis, Multiple Baseline Stereo, Hierarchical Fusion, Using More Cameras

Segmentation by Clustering: Segmentation, Human Vision: Grouping and Gestalt, Applications: Shot Boundary Detection and Background Subtraction, Image Segmentation by Clustering Pixels, Segmentation by Graph-Theoretic Techniques

Module-IV: **[10]**

Segmentation by Fitting a Model: The Hough Transform, Fitting Lines, Fitting Curves, Fitting as a Probabilistic Inference Problem

Tracking with Linear Dynamic Models Alignment: 2D and 3D, Problems of Analytical Euclidean Geometry, Camera Parameters and the Perspective Projection, Pose and Affine Projection Equations
Applications of Alignment: Last-Squares Parameter Estimation, Planning, Obstacle Avoidance and Control, Building Maps and Locating in them, Analytical Photogrammetry, An Application: Augmented Reality

Module-V: **[8]**
Introduction to Robotics: Social Implications of Robots, Brief history of Robotics, Attributes of Robots, Application areas of Robots and well as future potential, Robot Mechanisms Architectures, Attributes of Reactive Paradigm, Subsumption Architecture, Potential fields and perceptual servoing
Common sensing techniques for Reactive Robots: Logical sensors, Behavioural Sensor Fusion, Pro- prioceptive sensors, Proximity Sensors, Topological Planning and Metric Path Planning

TEXTBOOKS:

1. **David A. Forsyth & Jean Ponce**, *Computer Vision – A Modern Approach*, PHI Learning (Indian Edition), 2009.
2. **Robin Murphy**, *Introduction to AI Robotics*, MIT Press

REFERENCE BOOKS:

1. **E. R. Davies**, *Computer and Machine Vision*, Elsevier, 4th Edition, 2013.
2. **Maja J. Mataric**, *The Robotics Primer*, MIT Press.
3. **Richard Szeliski**, *Computer Vision: Algorithms and Applications*, Springer-Verlag, 2011.

24X6647: 5G & IOT TECHNOLOGIES

L	T	P	C
3	0	0	3

Course Outcomes: After Completion of the course, student should be able to

- Understand the application areas of IoT.
- Realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks.
- Apply building blocks of Internet of Things and characteristics.
- Analyse IOT and M2M.
- Identify end-points of physical devices

Module - I **[10]**

Overview of 5G Broadband Wireless Communications: Evolution of mobile technologies 1G to 4G (LTE, LTEA, LTEA Pro), An Overview of 5G requirements, Regulations for 5G, Spectrum Analysis and Sharing for 5G.

Module - II **[10]**

The 5G wireless Propagation Channels: Channel modelling requirements, propagation scenarios and challenges in the 5G modeling, Channel Models for mmWave MIMO Systems, 3GPP standards for 5G, IEEE 802.15.4

Module - III **[10]**

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, Iot Communication APIs, IoT enabled Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels and Templates, Domain Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle.

Module - IV **[8]**

IoT and M2M – Software defined networks, network function virtualization, difference between SDN and NFV for IoT. Basics of IoT System Management with NETCOZF, YANGNETCONF, YANG, SNMP NETOPEER

Module – V **[10]**

IoT Physical Devices and Endpoints - Introduction to Raspberry PI - Interfaces (serial, SPI, I2C). Programming – Python program with Raspberry PI with focus of interfacing external gadgets, controlling output, reading input from pins.

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. Jonathan Rodriguez, "Fundamentals of 5G Mobile Networks", John Wiley & Sons.
2. Amitabha Ghosh and Rapeepat Ratasuk "Essentials of LTE and LTE-A", Cambridge University Press.
3. Athanasios G. Kanatos, Konstantina S. Nikita, Panagiotis Mathiopoulos, "New Directions in Wireless Communication Systems from Mobile to 5G", CRC Press.
4. Theodore S. Rappaport, Robert W. Heath, Robert C. Danials, James N. Murdock "Millimeter Wave Wireless Communications", Prentice Hall Communications.

2476672: DEEP LEARNING LAB

L	T	P	C
0	0	2	1

Prerequisites: Machine Learning Lab

Course Objectives:

- To set up Python IDEs and deep learning libraries (Keras, TensorFlow, PyTorch) for experimentation.
- To implement basic deep learning programs including vector operations, regression models, and perceptrons.
- To design and train feedforward and convolutional neural networks for image processing tasks.
- To apply deep learning models for natural language processing tasks such as sentiment analysis using RNNs, LSTM, and GRU.
- To explore and implement autoencoder algorithms for data encoding and dimensionality reduction

Course Outcomes: After completion of the course, students should be able to

- Set up Python IDE and deep learning libraries (Keras, TensorFlow, PyTorch) for experiments.
- Implement basic deep learning programs like vector operations, regression, and perceptrons.
- Design feed-forward and convolutional neural networks for image tasks.
- Apply RNN, LSTM, and GRU models for natural language processing and sentiment analysis.
- Implement auto-encoders for data encoding and dimensionality reduction.

List of Experiments:

1. Setting up the Spyder IDE Environment and Executing a Python Program
2. Installing Keras, Tensor flow and Pytorch libraries and making use of them
3. Implement Simple Programs like vector addition in Tensor Flow.
4. Implement a simple problem like regression model in Keras.
5. Implement a perceptron in Tensor Flow / Keras Environment.
6. Implement a Feed-Forward Network in Tensor Flow/Keras.

7. Implement Convolution Neural Network on given data.
8. Applying the Convolution Neural Network on computer vision problems
9. Implement Recurrent Neural Network.
10. Apply convolutional neural network on speech recognition problem.
11. Image classification on MNIST dataset (CNN model with Fully connected layer)
12. Applying the Deep Learning Models in the field of Natural Language Processing
13. Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRUnotes
14. Applying the Auto encoder algorithms for encoding the real-world data

Text Books:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
2. The Elements of Statistical Learning by T. Hastie, R. Tibshirani, and J. Friedman, Springer.

References Books:

1. Bishop, C.M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.

2470580: FULLSTACK DEVELOPMENT LABORATORY

L	T	P	C
0	0	2	1

Pre-Requisites: Object Oriented Programming using Java Laboratory

Course Outcomes: After completion of course, student should be able to

- Design flexible and responsive Web applications using NodeJS, React, Express and Angular.
- Perform CRUD operations with MongoDB on huge amount of data.
- Apply NodeJS, ReactJS, Angular and MongoDB for developing responsive web applications.
- Develop real time applications using react components.
- Use various full stack modules to handle http requests and responses.

List of Experiments

1. Create an application to set up nodeJS environment and display "HelloWorld".
2. Create a NodeJS application for user login system.
3. Write a NodeJS 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. Createafooddeliverywebsitewhereuserscanorderfoodfromaparticularrestaurantlistedin 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 AngularJS .
11. Write an AngularJS 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 AngularJS.
13. Write a program to create a simple calculator Application using ReactJS.
14. Write a program to create a voting application using ReactJS

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.

TEXTBOOKS:

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.

REFERENCEBOOKS:

1. VasanSubramanian,ProMERNStack,FullStackWebAppDevelopmentwithMongo, Express, React, and Node, 2nd Edition, Apress, 2019.
2. ChrisNorthwood,TheFullStackDeveloper:YourEssentialGuidetotheEverydaySkills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. BradGreen&Seshadri.AngularJS.1stEdition.O'ReillyMedia,2013.
4. KirupaChinnathambi,LearningReact:AHands-OnGuidetoBuildingWebApplicationsUsing React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

2070025: HUMAN VALUES & PROFESSIONAL ETHICS

L	T	P	C
0	0	0	0

Prerequisite: Nil

Course Objectives:

- To make the students understand the types of roles they are expected to play in the society as practitioners of the civil engineering profession
- To develop some ideas of the legal and practical aspects of their profession.

Course Outcomes:

- To understand the importance of professional practice, Law and Ethics in their personal lives and professional careers.
- To learn the rights and responsibilities as an employee, team member and a global citizen

UNIT – I

Professional Practice and Ethics: Definition of Ethics, Professional Ethics - Engineering Ethics, Personal Ethics; Code of Ethics - Profession, Professionalism, Professional Responsibility, Conflict of Interest, Gift Vs Bribery, Environmental breaches, Negligence, Deficiencies in state-of-the-art; Vigil Mechanism, Whistle blowing, protected disclosures. Introduction to GST- Various Roles of Various Stake holders.

UNIT - II

Law of Contract: Nature of Contract and Essential elements of valid contract, Offer and Acceptance, Consideration, Capacity to contract and Free Consent, Legality of Object. Unlawful and illegal agreements, Contingent Contracts, Performance and discharge of Contracts, Remedies for breach of contract. **Contracts-II:** Indemnity and guarantee, Contract of Agency, Sale of goods Act -1930: General Principles, Conditions & Warranties, Performance of Contract of Sale.

UNIT – III

Arbitration, Conciliation and ADR (Alternative Dispute Resolution) system: Arbitration – meaning, scope and types – distinction between laws of 1940 and 1996; UNCITRAL model law – Arbitration and expert determination; Extent of judicial intervention; International commercial arbitration; Arbitration agreements – essential and kinds, validity, reference and interim measures by court; Arbitration tribunal appointment, challenge, jurisdiction of arbitral tribunal, powers, grounds of challenge, procedure and court assistance; Distinction between conciliation, negotiation, mediation and arbitration, confidentiality, resort to judicial proceedings, costs; Dispute Resolution Boards; Lok Adalats.

UNIT – IV

Engagement of Labour and Labour & other construction-related Laws: Role of Labour in Civil Engineering; Methods of engaging labour- on rolls, labour sub-contract, piece rate work; Industrial Disputes Act, 1947; Collective bargaining; Industrial Employment (Standing Orders) Act, 1946; Workmen's Compensation Act, 1923; Building & Other - Construction Workers (regulation of employment and conditions of service) Act (1996) and Rules (1998); RERA Act 2017, NBC 2017.

UNIT - V

Law relating to Intellectual property: Introduction – meaning of intellectual property, main forms of IP, Copyright, Trademarks, Patents and Designs, Secrets; Law relating to Copyright in India including Historical evolution of Copy Rights Act, 1957, Meaning of copyright – computer programs, Ownership of copyrights and assignment, Criteria of infringement, Piracy in Internet – Remedies and procedures in India; Law relating to Patents under Patents Act, 1970

TEXT BOOKS:

1. Professional Ethics: R. Subramanian, Oxford University Press, 2015.
2. Ravinder Kaur, Legal Aspects of Business, 4e, Cengage Learning, 2016.

REFERENCES:

1. Wadhera (2004), Intellectual Property Rights, Universal Law Publishing Co. RERA Act, 2017
2. T. Ramappa (2010), Intellectual Property Rights Law in India, Asia Law House.
3. O.P. Malhotra, Law of Industrial Disputes, N.M. Tripathi Publishers

IV YEAR II SEMESTER (VIII SEMESTER)

S.No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIE)	External (SEE)	Total
		Theory								
1		Professional Elective-V	PE	3	0	0	3	40	60	100
2		Professional Elective VI	PE	3	0	0	3	40	60	100
		Project								
1	2480589	Technical Seminar	PS	0	0	4	2	100	-	100
2	2480591	Project Stage–II	PS	0	0	18	9	40	60	100
Total Credits				**6**	**0**	**22**	**17**	**220**	**180**	**400**

24X6645: CYBER CRIME INVESTIGATION & DIGITAL FORENSICS

L T P C
3 0 0 3

Course Objectives:

- An introduction to the methodology and procedures associated with digital forensic analysis in a network environment.
- To understand the foundations of digital forensics, principles of digital evidence, and the role of computers in cybercrime.
- To explore digital investigation processes, methodologies, and scientific approaches for handling cybercrime scenes.
- To apply forensic techniques in analyzing digital evidence from violent crimes and reconstructing investigative scenarios.
- To analyze digital evidence from computer systems including Windows, Unix, and network environments.

Course Outcomes: After Completion of the course, students should be able to

- Understand the principles of digital forensics, digital evidence types, and cybercrime laws.
- Explain digital investigation models, crime scene handling, and preservation of digital evidence.
- Implement forensic procedures to extract, preserve, and analyze digital evidence from computers and networks.
- Analyze digital evidence in complex cases, including violent crimes, network incidents, and multi-system investigations.
- Evaluate digital forensic tools, methodologies, and legal frameworks to ensure admissible and reliable evidence.

MODULE – I [10]

Foundations of Digital Forensics: Digital Evidence, Principles of Digital Forensics, Challenging aspects of Digital Evidence, The Role of computers in crime, Cyber Crime Law.

MODULE – II [10]

Digital Investigations: Digital Investigation process models, Applying Scientific method in Digital Investigations, Handling a digital Crime scene: Fundamental Principles, Surveying and Preserving Digital Investigation.

MODULE – III [10]

Violent Crime and Digital Investigation: The role of Computers in violent crime, Processing Digital crime scene, Investigative Reconstruction, Digital Evidence as Alibi.

MODULE – IV [10]

Cyber stalking, Computer basics for Digital Forensics, Applying Forensics science to computers, Digital Evidence on windows systems, Digital Evidence on Unix systems.

MODULE – V [8]

Network Forensics: Networks basics for Digital Investigators, Applying Forensics science to networks, Digital Evidence on physical and data link layers, Digital Evidence on Network and Transport layers.

TEXT BOOK:

1. Digital Evidence and computer Crime by Eoghan Casey Academic Press Third Edition.

REFERENCE BOOKS:

1. Real Digital Forensics for Handheld Devices, E. P. Dorothy, Auerback Publications, 2013.

2. The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics,

3. J. Sammons, Syngress Publishing, 2012.

4. Handbook of Digital Forensics and Investigation, E. Casey, Academic Press, 2010.

24X0548: SOCIAL NETWORK ANALYSIS

L T P C
3 0 0 3

Course Objectives

- To understand the fundamentals of social media, social networks, and their collaborative technologies.
- To explore mechanisms for social network analysis, including measurement, mapping, and modeling of connections.
- To apply social network analysis tools such as NodeXL for data preparation, visualization, and metric computation.
- To analyze case studies of widely used social media platforms like email, Twitter, Facebook, YouTube, and Wikis.
- To evaluate patterns of content interaction and network prominence for research or decision-making in social media analytics.

Course Outcomes: After Completion of the course, students should be able to

- Understand the concepts of social media, social networks, and collaborative technologies.
- Explain methods for constructing and interpreting social network maps and metrics.
- Implement social network maps, data preparation, and visualization using NodeXL.
- Analyze interactions and connectivity patterns across platforms such as Twitter, Facebook, YouTube, and Wikis.
- Evaluate insights on content flow, network influence, and collaborative patterns for social media applications.

MODULE – I [10]

Introduction: Social Media and Social Networks Social Media: New Technologies of Collaboration Social Network Analysis: Measuring, Mapping, and Modelling collections of Connections.

MODULE – II [10]

NodeXL, Layout, Visual Design, and Labelling, Calculating and Visualising Network Metrics, Preparing Data and Filtering, Clustering and Grouping.

MODULE – III [10]

CASE STUDIES: Email: The lifeblood of Modern Communication. Thread Networks: Mapping Message Boards and Email Lists Twitter: Conversation, Entertainment and Information

MODULE – IV [10]
CASE STUDIES: Visualizing and Interpreting Facebook Networks, WWW Hyperlink Networks

MODULE – V [8]
CASE STUDIES: You Tube: Contrasting Patterns of Content Interaction, and Prominence. Wiki Networks: Connections of Creativity and Collaboration

TEXT BOOK:

1. Hansen, Derek, Ben Sheiderman, Marc Smith, Analyzing Social Media Networks with NodeXL: Insights from a Connected World, Morgan Kaufmann, 2011.

REFERENCE BOOKS:

1. Avinash Kaushik, Web Analytics 2.0: The Art of Online Accountability, Sybex, 2009.

2. Marshall Sponder, Social Media Analytics: Effective Tools for Building, Interpreting and Using Metrics, 1st Edition, MGH, 2011.

24X0549: QUANTUM COMPUTING

B.Tech. IV 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
- To understand the mathematical, physical, and biological foundations necessary for quantum computation.
- To apply knowledge of qubits, quantum gates, and circuits for representing quantum information.
- To analyze major quantum algorithms and their relationship with classical computational complexity.

Course Outcomes After completion of this course, students should be able to

- Describe the basics of quantum computing, including differences between classical and quantum operations.
- Explain the physical implementation of qubits and their role as units of quantum information.
- Implement selected quantum algorithms and construct simple quantum circuits for problem solving.
- Analyze the efficiency and significance of major quantum algorithms in comparison with classical approaches.
- Evaluate the impact of quantum computing on cryptography, including strengths and limitations of quantum protocols.

MODULE – I [10]

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

MODULE – II [10]

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)

MODULE – III [10]

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.

MODULE – IV [10]

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.

MODULE – V [8]

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

TEXT BOOK:

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.

24X6652: INDUSTRIAL IOT

L T P C
3 0 0 3

Course Objectives:

- To provide students with a good depth of knowledge of Designing Industrial IOT Systems for various applications.
- To understand the fundamentals of Industrial Internet, key IIoT technologies, and their applications in industry and retail.
- To explore the role of cyber-physical systems, IP mobility, SDN, cloud/fog computing, big data, AI, and machine learning in IIoT.
- To apply IIoT reference architectures, three-tier topologies, and advanced data analytics for industrial applications.
- To analyze industrial communication protocols, including legacy and modern protocols, wireless technologies, and gateways for IIoT systems.

Course Outcomes: After completion of the course, students should be able to

- Identify the Key opportunities and benefits in Industrial IoT
- Apply virtual network to demonstrate the use of Cloud in Industrial IoT
- Analyze industrial IoT Three tier topology and data management system
- Summarize Legacy Industrial and Modern Communication Protocols
- Describe Middleware Architecture, LoRaWAN- and Augmented reality

MODULE – I [10]

Introduction To Industrial Internet And Use-Cases: Industrial Internet- Key IIoT TechnologiesInnovation and the IIoT -Key Opportunities and Benefits -The Digital and Human Workforce – Logistics and the Industrial Internet- IOT Innovations in Retail.

MODULE – II [10]

The Technical And Business Innovators Of The Industrial Internet: Cyber Physical Systems (CPS) – IP Mobility – Network Virtualization - SDN (Software Defined Networks)- The Cloud and Fog – Role of Big Data in IIOT - Role of Machine learning and AI in IIOT

MODULE – III [10]

IIOT Reference Architecture: Industrial Internet Architecture Framework (IIAF) -Industrial Internet Viewpoints -. Architectural Topology: The

Three-Tier Topology- Key System Characteristics- Data Management- Advanced data analytics.

MODULE – IV [10]

Protocols for Industrial Internet Systems: Legacy Industrial Protocols - Modern Communication Protocols-Proximity Network Communication Protocols- Wireless Communication TechnologiesGateways: industrial gateways - CoAP (Constrained Application Protocol)- NFC.

MODULE – V [8]

Middleware Software Patterns and IIOT Platforms: Publish/Subscribe Pattern: MQTT, XMPP, AMQP, DDS- Middleware Architecture- SigFox- LoRaWAN Augmented reality- Real-World Smart Factories

Application of IIOT: Case study: Health monitoring, Iot smart city, Smart irrigation, Robot surveillance.

TEXT BOOKS:

1. Gilchrist, Alasdair, “Industry 4.0 The Industrial Internet of Things”, Apress, 2017.

2. Zaigham Mahmood, “The Internet of Things in the Industrial Sector: Security and Device connectivity, smart environments and Industry 4.0 (Springer), 2019.

REFERENCE BOOKS:

1. Sabina Jeschke, Christian Brecher, Houbing Song, Danda B. Rawat “Industrial Internet of Things: Cyber manufacturing Systems” (Springer), 2017.

2. Industrial IoT Challenges, Design Principles, Applications, and Security by Ismail Butun (editor)

3. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014.

4. Michahelles, “Architecting the Internet of Things”, ISBN 978-3- 642-19156-5 e-ISBN 978-3-642- 19157-2, Springer

5. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 20132 Cuno Pfister, Getting Started with the Internet of Things, O ‟Reilly Media, 2011, ISBN: 978-1-4493-9357-1

E-BOOKS

1. https://www.apress.com/gp/book/9781484220467

CLOUD SECURITY

Course Outcomes: After completion of the course, students should be able to

- Understand the fundamental concepts of distributed systems, clustering, virtualization, and cloud computing models
- Analyze Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) by evaluating virtual machine provisioning, migration techniques, cluster-based cloud enhancement, and secure distributed data storage platforms.
- Apply cloud security principles by demonstrating data protection techniques (data at rest, in transit, and in use), data lifecycle management, cloud data auditing, and key management practices in AWS and Azure environments.
- Evaluate Identity and Access Management (IAM) and Federated Identity Management systems, and perform IAM audits using AWS IAM tools and cloud portals to ensure secure access governance.
- Design and implement cloud-based solutions by integrating virtualization, storage services, security controls, and IAM policies, and demonstrate practical skills using AWS CLI, Azure Portal, and workflow engines for cloud applications.

MOUDEL-I 10

FUNDAMENTALS: System Modeling, Clustering and Virtualization: distributed system models and enabling technologies, computer clusters for scalable parallel computing, virtual machines and virtualization of clusters and data centers. Introduction to cloud computing, migrating into cloud, enriching the integration of service paradigm for cloud era, the enterprise cloud computing paradigm.

MOUDEL-II 8

INFRA STRUCTURE AS SERVICE (IAAS) & PLATFORM AND SOFTWARE SERVICE(PAAS/SAAS): Virtual machine provisioning and migration services, on the management of virtual machines for cloud infrastructure, enhancing cloud computing environments using a cluster as service, secure distributed data storage in cloud computing Aneka, comet cloud, T-systems, work flow engine for clouds ,understanding scientific applications for cloud environments.

MOUDEL-III 8

Data Protection (rest, at transit, in use), Data Information lifecycle, Cloud Data Audit (Intro, Audit, Best Practice): Aws - EBS, S3, Azure - SAS, Demo-Aws cli & PowerShell & Amazon, Azure portal, Key management, Cloud Key management Audit (Introduction, Audit, Best Practice): AWS -KMS, Azure - Azure Key Vault

MOUDEL-IV 8

Introduction to Identity and Access Management, Introduction to Federated Identity Management, Case Study, Cloud IAM Audit (Intro, Audit, Best Practice): Aws -IAM, Demo – AWS CLI & Amazon portal

MOUDEL-V 8

Cloud Application Challenges, OWSAP Top 10, SECURE SDLC, DevsecOps, Cloud Trail, Cloud watch, Lambda

TEXT BOOKS

Ronald L. Krutz, Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, 30 July 2010
Securing the Cloud: Cloud Computer Security Techniques and Tactics – Illustrated, 1 June 2011

24X0530: NATURAL LANGUAGE PROCESSING

L	T	P	C
3	0	0	3

Prerequisites: Data structures, compiler design

Course Objectives:

- To understand the structure of words and documents, and explore morphological analysis techniques.
- To study syntax and parsing of natural language using data-driven approaches and treebanks.
- To address ambiguity resolution in parsing and analyze multilingual syntax issues.
- To explore semantic parsing, predicate-argument structures, and meaning representation systems.
- To learn language modeling techniques including n-gram models, Bayesian estimation, and multilingual adaptations.

Course Outcomes: After completion of the course, students should be able to

- Analyze the structure of words and documents using morphological models and feature extraction techniques.
- Apply parsing algorithms and treebank approaches for syntactic analysis of natural language.
- Resolve ambiguity in parsing and handle multilingual syntax challenges.
- Develop semantic parsing models and represent meaning using predicate-argument structures.
- Build language models using n-grams, Bayesian methods, and multilingual adaptations.

Module – I **[10]**

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

Module – II **[10]**

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

Module – III **[10]**

Syntax II: Models for Ambiguity Resolution in Parsing, Multilingual Issues

Semantic Parsing I: Introduction, Semantic Interpretation, System Paradigms, Word Sense

Module – IV **[8]**

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

Module – V **[10]**

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

Text Books:

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

Reference Books:

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.

24X0542: COGNITIVE COMPUTING

L	T	P	C
3	0	0	3

Course Objectives:

- To understand the fundamentals of cognitive science and cognition modeling paradigms, including logic-based, connectionist, and Bayesian approaches.
- To study computational models of memory and language, including episodic and semantic memory and psycholinguistics.
- To model interactions between language, memory, and learning, and analyze classical models of rationality, reasoning, and decision making.
- To explore formal models of generalization, causality, categorization, analogy, and cognitive development, along with cognitive architectures like ACT-R and SOAR.
- To understand cognitive computing architectures (DeepQA, UIMA), structured knowledge, and the design of cognitive applications with real-world business implications.

Course Outcomes: After completion of the course, students should be able to

- Analyze cognitive science concepts and apply various cognition modeling paradigms.
- Implement computational models for memory, language, and psycholinguistic tasks.
- Model interactions between language, memory, learning, and decision-making processes.
- Apply formal models of generalization, analogy, and cognitive development using architectures like ACT-R and SOAR.
- Design cognitive computing applications using architectures such as DeepQA and UIMA.

Module – I **[10]**

Introduction to Cognitive Science: Understanding Cognition, IBM's Watson, Design for Human Cognition, Augmented Intelligence, Cognition Modeling Paradigms: Declarative/ logic-based computational cognitive modeling, connectionist models of cognition, Bayesian models of cognition, a dynamical systems approach to cognition.

Module – II **[8]**

Cognitive Models of memory and language, computational models of episodic and semantic memory, modeling psycholinguistics.

Module – III **[10]**

Cognitive Modeling: modeling the interaction of language, memory and learning, Modeling select aspects of cognition classical models of rationality, symbolic reasoning and decision making.

Module – IV **[10]**

Formal models of inductive generalization, causality, categorization and similarity, the role of analogy in problem solving, Cognitive Development Child concept acquisition. Cognition and Artificial cognitive architectures such as ACT-R, SOAR, OpenCog, CopyCat, Memory Networks.

Module – V **[10]**

DeepQA Architecture, Unstructured Information Management Architecture (UIMA), Structured Knowledge, Business Implications, Building Cognitive Applications, Application of Cognitive Computing and Systems.

Text Books:

1. The Cambridge Handbook of Computational Psychology by Ron Sun (ed.), Cambridge University Press.

Reference Books:

1. Judith S. Hurwitz, Marcia Kaufman, Adrian Bowles Cognitive Computing and Big Data Analytics, Wiley
2. Vijay V Raghavan, Venkat N. Gudivada, Venu Govindaraju, Cognitive Computing: Theory and Applications: Volume 35 (

Generative AI

Course Outcomes: After completion of the course, student should be able to

- Understand the Fundamentals of Generative AI
- Identify the performance of generative models using metrics
- Analyze the ethical implications of generative AI
- Apply algorithms to build and train generative models using frameworks
- Compare the performance of various generative AI architectures

Module-1 8

Introduction to AI and Machine Learning-Types of Generative Models (e.g., LLM,SLM,GANs, VAEs, Autoregressive Models)- Neural Networks: Basic Architecture, Backpropagation, Activation Functions-Deep Learning Basics and its Applications-Unsupervised vs. Supervised Learning

Module-2 10

Introduction to Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs)- Understanding the Generator-Discriminator Architecture in GANs-Latent Space Representation and Loss Functions-Training Strategies, Optimization, and Hyperparameter Tuning-Applications and Case Studies in Image, Video, and Text Generation-Diffusion Models and their Use in Modern AI Art Generation

Module-3 9

Introduction to TensorFlow and PyTorch for Generative AI Building GANs and VAEs from Scratch Hands-on Projects: Generating Images, Music, and Text - Model Evaluation Techniques (FID Score, Inception Score, BLEU Score) - Fine-tuning Pretrained Models for Specialized Applications

Module-4 10

Ethical Challenges in AI: Bias, Fairness, and Accountability- Deepfakes and Synthetic Media: Risks and Regulations - AI in Content Creation: Copyright, Ownership, and Creativity - Data Privacy in Generative AI Systems - Bias and Fairness in Training Data: Identifying and Mitigating

Module-5 9

AI in Digital Art and Content Creation - Music Generation Using Neural Networks - Healthcare Applications: Drug Discovery and Medical Imaging - Natural Language Generation (NLG) and Chatbots - Case Studies: Generative AI in Gaming, Fashion, and Virtual Reality

Text Book:

1. Gohil, P. (2019). Machine learning with Tensor Flow. BPB Publications.
2. Akerkar, R. (2020). Deep learning: A practitioner's approach. Springer.
3. Arora, R. (2021). Artificial intelligence: A guide for thinking humans. Wiley India Pvt. Ltd.
4. Schwab, K. (2017). The fourth industrial revolution (Indian edition). Penguin Random House India.