GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

(Autonomous)

(Approved by AICTE, New Delhi and Permanently Affiliated to Andhra University, Visakhapatnam)

Madhurawada :: Visakhapatnam – 530 048

COURSE STRUCTURE

(Applicable for the Academic Year 2024-25 onwards)

DEPARTMENT OF INFORMATION TECHNOLOGY

B.Tech. Information Technology

SCHEME AND SYLLABI

(With effect from 2024-25 admitted batch)

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN
(Autonomous)
(Approved by AICTE, New Delhi and Permanently Affiliated to Andhra University, Visakhapatnam)
Madhurawada :: Visakhapatnam – 530 048

COURSE STRUCTURE

(Applicable for the Academic Year 2024-25 onwards)
DEPARTMENT OF INFORMATION TECHNOLOGY

B.Tech. Information Technology
SCHEME AND SYLLABI
(With effect from 2024-25 admitted batch)

I Year - I Semester								
Course Code	**Category**	**Course Title**	**Hours per Week**		**Internal Marks**	**External Marks**	**Total Marks**	**Credits**
			L	**P**				
24BM11RC01	BS	Calculus and Differential Equations	3	0	30	70	100	3
24BC11RC01	BS	Green Chemistry	3	0	30	70	100	3
24HE11RC01	HSS	English	3	0	30	70	100	3
24CT11RC02	ES	Problem Solving using C	3	0	30	70	100	3
24CT11RC01	ES	Fundamentals of Computers	3	0	30	70	100	3
24HE11RC02	HSS	Communication Skills Lab	0	3	50	50	100	1.5
24CT11RC04	ES	Problem Solving using C Lab	0	3	50	50	100	1.5
24CT11RC03	ES	Computer Engineering Workshop	0	3	50	50	100	1.5
							Total Credits	**19.5**

I Year - II Semester								
Course Code	**Category**	**Course Title**	**Hours per Week**		**Internal Marks**	**External Marks**	**Total Marks**	**Credits**
			L	**P**				
24BM11RC02	BS	Linear Algebra and Vector Calculus	3	0	30	70	100	3
24BP11RC01	BS	Engineering Physics	3	0	30	70	100	3
24CT11RC06	ES	Python Programming	3	0	30	70	100	3
24EC11RC04	ES	Elements of Electronics Engineering	3	0	30	70	100	3
24EC11RC05	ES	Digital Logic Design	3	0	30	70	100	3
24BP11RC02	BS	Engineering Physics Lab	0	3	50	50	100	1.5
24CT11RC07	ES	Python Programming Lab	0	3	50	50	100	1.5
24CT11RC08	ES	Web Technologies Fundamentals Lab	0	3	50	50	100	1.5
							Total Credits	**19.5**

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN
(Autonomous)

(Approved by AICTE, New Delhi and Permanently Affiliated to Andhra University, Visakhapatnam)

Madhurawada :: Visakhapatnam – 530 048

COURSE STRUCTURE

(Applicable for the Academic Year 2024-25 onwards)

DEPARTMENT OF INFORMATION TECHNOLOGY

B.Tech. Information Technology

SCHEME AND SYLLABI

(With effect from 2024-25 admitted batch)

II Year - I Semester

Course Code	Category	Course Title	Hours per Week		Internal Marks	External Marks	Total Marks	Credits
			L	P				
24BM11RC05	ES	Discrete Mathematical Structures	3	0	30	70	100	3
24CT11RC11	PC	Computer Organization	3	0	30	70	100	3
24CT11RC09	PC	Data Structures	3	0	30	70	100	3
24CT11RC10	PC	Object Oriented Programming through Java	3	0	30	70	100	3
24CT11RC13	PC	Operating Systems	3	0	30	70	100	3
24CT11RC14	PC	Data Structures Lab	0	3	50	50	100	1.5
24CT11RC15	PC	Object Oriented Programming through Java Lab	0	3	50	50	100	1.5
24CT11RC17	PC	Operating Systems Lab	0	3	50	50	100	1.5
24CT11SC01	SC	Continuous Integration and Continuous Delivery using DevOps	1	2	50	50	100	2
24BC11MC01	MC	Environmental Science	2	0	-	100	100	0
							Total Credits	**21.5**

II Year - II Semester

Course Code	Category	Course Title	Hours per Week		Internal Marks	External Marks	Total Marks	Credits
			L	P				
24BM11RC06	BS	Probability and Statistics	3	0	30	70	100	3
24HM11RC01	HSS	Managerial Economics	3	0	30	70	100	3
24CT11RC18	PC	Design and Analysis of Algorithms	3	0	30	70	100	3
24CT11RC12	PC	Database Management Systems	3	0	30	70	100	3
24CT11RC20	PC	Formal Languages and Automata Theory	3	0	30	70	100	3
24CT11RC21	PC	Algorithms Lab through C++	0	3	50	50	100	1.5
24CT11RC16	PC	Database Management Systems Lab	0	3	50	50	100	1.5
24IT11SC01	SC	R Programming	1	2	50	50	100	2
24HM11MC01	MC	Professional Ethics and Human Values	2	0	-	100	100	0
24IT11SW01	MC	Social Activity/Life Skills	0	0	-	-	-	0
							Total Credits	**20**

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN
(Autonomous)

(Approved by AICTE, New Delhi and Permanently Affiliated to Andhra University, Visakhapatnam)

Madhurawada :: Visakhapatnam – 530 048

COURSE STRUCTURE

(Applicable for the Academic Year 2024-25 onwards)

DEPARTMENT OF INFORMATION TECHNOLOGY

B.Tech. Information Technology

SCHEME AND SYLLABI

(With effect from 2024-25 admitted batch)

III Year - I Semester

Course Code	Category	Course Title	Hours per Week		Internal Marks	External Marks	Total Marks	Credits
			L	P				
24CT11RC22	PC	Computer Networks	3	0	30	70	100	3
24CT11RC19	PC	Data Warehousing & Data Mining	3	0	30	70	100	3
24IA11RC01	PC	Artificial Intelligence	3	0	30	70	100	3
	PE	Professional Elective-1	3	0	30	70	100	3
	OE	Open Elective-1	3	0	30	70	100	3
24CT11RC24	PC	Computer Networks Lab	0	3	50	50	100	1.5
24CI11RC01	PC	Data Mining Lab	0	3	50	50	100	1.5
24CT11SC02	SC	Skill Course :MERN Stack Development	1	2	50	50	100	2
24IT11IN01	INT	Internship-I			-	100	100	2
24IT11MC01	MC	Design Thinking, Innovation & Entrepreneurship	2	0	50	50	100	0
							Total Credits	**22**

III Year - II Semester

Course Code	Category	Course Title	Hours per Week		Internal Marks	External Marks	Total Marks	Credits
			L	P				
24CT11RC23	PC	Machine Learning	3	0	30	70	100	3
24CI11RC02	PC	Cryptography and Network Security	3	0	30	70	100	3
24CI11RC03	PC	Object Oriented Software Engineering	3	0	30	70	100	3
	PE	Professional Elective-2/MOOCS Course	3	0	30	70	100	3
	OE	Open Elective-2	3	0	30	70	100	3
24CT11RC25	PC	Machine Learning Lab	0	3	50	50	100	1.5
24CI11RC04	PC	Cryptography and Network Security Lab	0	3	50	50	100	1.5
24CI11RC05	PC	Object Oriented Software Engineering Lab	0	3	50	50	100	1.5
24HE11SC01	SC	Soft Skills	1	2	50	50	100	2
24HM11MC02	MC	IPR & Patents	2	0	-	100	100	0
							Total Credits	**21.5**

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

(Autonomous)

(Approved by AICTE, New Delhi and Permanently Affiliated to Andhra University, Visakhapatnam)

Madhurawada :: Visakhapatnam – 530 048

COURSE STRUCTURE

(Applicable for the Academic Year 2024-25 onwards)

DEPARTMENT OF INFORMATION TECHNOLOGY

B.Tech. Information Technology

SCHEME AND SYLLABI

(With effect from 2024-25 admitted batch)

IV Year - I Semester								
Course Code	**Category**	**Course Title**	**Hours per Week**		**Internal Marks**	**External Marks**	**Total Marks**	**Credits**
			L	**P**				
	HSS	HSS Elective	3	0	30	70	100	3
	PC	Professional Elective-3	3	0	30	70	100	3
	PC	Professional Elective-4	3	0	30	70	100	3
	PC	Professional Elective-5	3	0	30	70	100	3
	OE	Open Elective-3	3	0	30	70	100	3
	OE	Open Elective-4	3	0	30	70	100	3
24CI11SC01	SC	Deep Learning Essentials with Python	1	2	50	50	100	2
24IT11IN02	INT	Internship II			-	100	100	2
							Total Credits	**22**
IV Year - II Semester								
Course Code	**Category**	**Course Title**			**Internal Marks**	**External Marks**	**Total Marks**	**Credits**
24IT11PR01	PROJ	Project Work			100	100	200	14
							Total Credits	**14**

Professional Electives in IT		
SNO	COURSE CODE	SUBJECT
1	24CT11PE01	API and Microservices
2	24CI11PE01	Block chain Technology
3	24IT11PE01	Cloud Computing
4	24CI11PE02	Deep Learning
5	24CI11PE03	Distributed Systems
6	24IA11PE01	Data Science.
7	24CT11PE02	Ethical Hacking
8	24CT11PE03	Generative AI Models
9	24IA11PE02	Principles of Complier Design
10	24IT11PE02	Image Processing
11	24IT11PE03	Information Retrieval and Systems
12	24CI11PE04	Hadoop and Big Data Analytics
13	24CT11PE04	Mobile Application Development
14	24IA11PE03	Natural Language Processing (NLP)
15	24CT11PE05	Soft Computing
16	24IT11PE04	Unix and Shell Programming

Open Electives offered by IT	
24IT11EL01	Introduction to Blockchain Technology
24IT11EL02	Introduction to Cloud Computing.
24IT11EL03	Information Retrieval Search Engines

B.Tech. Minor in IT	
24IT11MN01	Computer Networks
24IT11MN02	Cryptography and Network Security
24IT11MN03	Cyber Security & Forensics
24IT11MN04	Ethical Hacking
24IT11MN05	Unix & Shell Programming
24IT11MN06	Computer Networks Lab
24IT11MN07	Linux Programming Lab

B.Tech. Honors in IT	
24CT11HN01	Quantum Computing
24CT11HN02	Social Media Analytics
24CI11HN01	Fundamentals of Reinforcement Learning
24IT11HN01	Information Security Analysis and Audit
24IT11HN02	Malware Analysis
24CI11HN02	Software Metrics

- MOOCs course must be of minimum 12 weeks duration in any of the courses approved by the department from time to time and are advanced for both Minors and Honors

CALCULUS AND DIFFERENTIAL EQUATIONS

I Year B. Tech. I Semester

[Common to EEE, ECE, CSE, IT, CSE (AI&ML)]

L	T	P	C
3	0	0	3

Course Code: 24BM11RC01

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

CO1: Examine the functional dependency and utilize Taylor's theorem to expand the function of two variables. (L3)

CO2: Determine extrema of multivariable functions. (L5)

CO3: Utilize double and triple integrals to evaluate areas of plane curves and the volumes of solids. (L5)

CO4: Solve the first order differential equations and higher order differential equations with constant coefficients, apply the techniques to solve problems related to various engineering fields. (L3)

CO5: Find the Laplace Transforms of various functions and apply it to solve ordinary differential Equations with initial conditions. (L3)

UNIT-I: **8 Lectures**

Partial Differentiation: Functions of two or more variables: Introduction - Partial derivatives - Total derivative - Change of variables - Jacobians – Functional dependence - Taylor's theorem for function of two variables. [Sections: 5.1, 5.2, 5.5, 5.6, 5.7, 5.9 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Examine the functional dependencies using the Jacobian. (L3)
2. Make use of Taylor's theorem to write series expansion of function of two variables. (L3)

UNIT-II: **7 Lectures**

Applications of Partial Differentiation: Maxima and minima of function of two and three variables - Constrained maximum/minimum problems using Lagrange's method of multipliers. [Sections: 5.11, 5.12 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Determine the critical points, maxima and minima of functions several variables. (L5)

UNIT-III: **11 Lectures**

Integral Calculus: Introduction - Double integrals - Change of order of integration - Double integrals in polar Coordinates - Triple integrals - Change of variables.

Applications: Area enclosed by plane curves - Volumes of solids. [Sections: 7.1 – 7.7 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Evaluate double integrals and triple integrals in Cartesian and polar coordinates also over the given region. (L5)
2. Evaluate the double integrals by change of order of integration. (L5)
3. Evaluate double and triple integrals by change of variables. (L5)
4. Determine the areas and volumes using multiple integrals. (L5)

UNIT-IV: **11 Lectures**

Differential Equations and its Applications: Exact differential equations - Equations reducible to exact equations - Solutions of higher order linear ordinary differential equations with constant coefficients - Method of variation of parameters.

Applications: Orthogonal Trajectories - Newton's law of cooling - Law of Natural growth and decay. [Sections: 11.11, 11.12, 12.3, 12.6, 12.8, 13.1 – 13.8 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Solve the first-order differential equations and higher order Linear differential equations with constant coefficients. (L3)
2. Apply the techniques to solve problems related to various engineering fields. (L3)

UNIT-V: **11 Lectures**

Laplace Transforms: Introduction - Existence conditions - Transforms of elementary functions - Properties of Laplace transforms - Transforms of derivatives - Transforms of integrals - Multiplication by t - Division by t– Evaluation of integrals by Laplace transforms - Laplace transforms of Unit step function, Unit impulse function and Laplace transforms of periodic functions - Inverse Laplace transforms - Convolution theorem - Second shifting theorem.

Applications: Laplace Transforms to ordinary differential Equations. [Sections: 21.1 – 21.5, 21.7 – 21.15, 21.17, 21.18 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Find the Laplace transform of the function. (L1)
2. Find the Inverse Laplace transform of a function. (L1)
3. Make use of convolution theorem to find the Inverse Laplace transform of a function. (L3)
4. Apply Laplace transform to solve ordinary differential equations. (L3)

Text Books:

1. B. S. Grewal, Higher Engineering Mathematics, 45th Edition, Khanna Publishers, 2024.

Reference Books:

1. R K Jain and SRK Iyengar, Advanced Engineering Mathematics, Narosa Publishers, 5th Edition, 2016.
2. Erwin Kreyszig, Advanced Engineering Mathematics, 10th edition, John Wiley & Sons, 2011.
3. Peter V. O'Neil, Advanced Engineering Mathematics, 7th edition, Cengage Learning, 2011.
4. Greenberg, Advanced Engineering Mathematics, 2nd Edition, Pearson, 2017.

Web References:

1. https://nptel.ac.in/courses/111104125
2. https://nptel.ac.in/courses/111105160

GREEN CHEMISTRY

I Year B. Tech. I semester
[Common to CSE, EEE, IT]

L	T	P	C
3	0	0	3

Course Code: 24BC11RC01

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

CO1: To develop knowledge about water and its treatment for industrial and potable purpose. (L3)

CO2: Utilize the theory of construction and discharge reactions of various types of batteries are used in commercial society. (L3)

CO3: Explain the importance of working principle, fabrication of electrodes and other components, advantages, disadvantages and environmental aspects of fuel cells. (L4)

CO4: Classify the corrosion mechanism of metals and factors influenced by rate and extent of corrosion and categorize the reasons for corrosion control methods. (L4)

CO5: Apply green chemistry technology processes the knowledge for solving existing challenges faced in various engineering and societal areas. (L5)

UNIT-I **10 Lectures**

Water-Technology: Sources of Water – Impurities and their influence of living systems – WHO Limits – Hardness and its Determination – Boiler Troubles and their removal – Water Softening Methods – Lime-Soda, Zeolite and Ion Exchange - Municipal Water Treatment-Break Point Chlorination – Desalination of Sea Water –Reverse Osmosis Method Electro-dialysis. Chemical analysis of water. [TB1: Chapter1]

Learning Outcomes:

At the end of the unit the student will be able to

1. Explain the principles of reverse osmosis (L4)
2. Compare the quality of drinking water with BIS and WHO standards (L4)
3. Illustrate problems associated with Boiler Troubles (L2)
4. Demonstrate the estimation of hardness of water (L3)

UNIT-II **10 Lectures**

Batteries: Primary batteries: The Chemistry-Types: Zinc-carbon (Leclanche type), zinc alkaline (Duracell), zinc/air batteries; Lithium primary cells and lithium-ferrous sulphide cells.

Secondary batteries: Lead acid and VRLA (valve regulated (sealed) lead acid), nickel cadmium, nickel-

zinc, nickel- metal hydride batteries, lithium-ion batteries, ultrathin lithium polymer cells. Advanced Batteries for electric vehicles, requirements of the battery – sodium-beta and redox batteries. [TB1: Chapter6]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Deduct the working mechanism of various types of cells (L5)
2. Illustrate difference between primary and secondary cells (L2)
3. List the environmental applications of Various types of batteries (L4)
4. Utilize the manufacturing methods of advanced batteries for electric vehicles (L3)

UNIT-III **8 Lectures**

Fuel Cells: Description, working principle, anodic, cathodic and cell reactions, fabrication of electrodes and other components, applications, advantages, disadvantages and environmental aspects of the following types of fuel cells: Proton Exchange Membrane Fuel Cells, alkaline fuel cells, phosphoric acid, solid oxide, molten carbonate, direct methanol fuel cells- Membranes and Fuels, Electrochemical Sensors and Optical fibres. [TB1: Chapter 6]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Explain the fundamental theories of fuel cells (L2)
2. Classify types of fuel cells (L4)
3. Make use of the various components fabrication of fuel cells (L3)
4. Distinguish the advantages, disadvantages and environmental aspects of fuel cells(L4)

UNIT-IV **10 Lectures**

Corrosion: Origin and Theory – Types of Corrosion: Chemical and Electrochemical; Pitting, inter granular, Waterline, Stress – Galvanic Series – Factors Effecting Corrosion. Corrosion Controlling Methods, Protective Coatings, Electroplating and Electroless Plating, Paints, Varnishes, Lacquers, Enamels. [TB1: Chapter 7]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Explain the Mechanism of corrosion (L4)
2. List the various types of Corrosion (L4)
3. Describe the Factors Effecting Corrosion (L5)
4. Differentiate between Paints Lacquers Enamels (L4)

UNIT-V **10 Lectures**

Green-Chemistry and Technology: Introduction and significance of green chemistry, Goals of green chemistry, 12 principles of green chemistry, toxicity of chemicals, concept of zero pollution technologies; Aqueous phase method, Super critical fluid method, Phase transfer catalytic method, atom economy, functional toxicity vs non- functional toxicity, functional group approaches to green chemistry, Elimination of toxic functional group, optimization of frameworks for the design of greener synthetic pathways -Green synthesis of Adipic acid and Paracetamol- energy minimization-Microwave synthesis, ultra sound assisted method Bio catalyzed reaction and Only explanation with examples Processes involving solid catalysts – zeolites, ion exchange resins, applications of Green Chemistry, Green solvents, green fuels & propellants. [TB2: Chapter- 2,3,4&5]

Learning Outcomes:

At the end of the unit the student will be able to

1. Processes involving Green Chemistry and apply the knowledge for solving existing
2. Challenges faced in various engineering and societal areas (L-5)
3. Differentiate between functional toxicity vs non- functional toxicity (L-4)
4. Explain the green chemistry, 12 principles (L-2)

Textbooks:

1. Engineering Chemistry – PC Jain and M. Jain –15th Edition, Dhanpath Rai and Sons, New Delhi.
2. Anastas, P. T., Warner, J. C. Green Chemistry: Theory and Practice, Oxford UniversityPress Inc., New York, 1998.

Reference Books:

1. M. Aulice Scibioh and B. Viswanathan ‘Fuel Cells – principles and applications’, University Press India (2006).
2. A Textbook of Engineering Chemistry – S. S. Dara – S. Chand & Co. New Delhi
3. Handbook of Green Chemistry and Technology; by James Clarke and Duncan Macquarrie; Blakwell Publishing.

Web References:

1. https://archive.nptel.ac.in/course.html
2. https://nptel.ac.in/courses/engineering

ENGLISH

I Year B. Tech. I Semester

[Common to CSE, IT]

L	T	P	C
3	0	0	3

Course Code: 24HE11RC01

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

CO1: Develop moral values and inner peace(L3), Demonstrate the use of LSRW skills, Vocabulary and basic grammar concepts. (L3)

CO2: Identify the impact of different social backgrounds (L3), recognize the poet's insights. (L4) Demonstrate the use of LSRW skills, Vocabulary and basic grammar concepts. (L3)

CO3: Analyse socio-cultural context(L4), Establish effective interpersonal and communication skills(L3), Demonstrate the use of LSRW skills, Vocabulary and basic grammar concepts(L3)

CO4: Focus on value of education(L4), Relate to the poet's nostalgia(L4), Demonstrate the use of LSRW skills, Vocabulary and basic grammar concepts (L3)

CO5: Determine the central idea of the text(L3), Identify one's resilience(L3), Demonstrate the use of LSRW skills, Vocabulary and basic grammar concepts (L3)

UNIT-I **10 Lectures**

Prose:	Swami Vivekananda: The Secret of Work
Poetry:	Grenville Kleiser: Stay Calm
Grammar & Vocabulary:	Synonyms & Antonyms
Listening:	Listening for Context and Specific Information
Speaking:	Introducing Oneself and Others
Writing:	Basics of writing

Learning Outcomes:

At the end of the unit, the student will be able to

1. Interpret the spiritual growth and capacity building of the individual in the 21st century. (L3)
2. Use appropriate synonyms and antonyms to communicate effectively. (L3)
3. Listen and understand for specific information in the audio(L2)
4. Establish connections between oneself and others(L3)
5. Apply appropriate punctuation marks for clarity and organization of written text (L3)

UNIT-II **10 Lectures**

Prose:	Katherine Mansfield: The Doll's House
Poetry: Rabindranath Tagore:	Where the Mind Is Without Fear
Grammar & Vocabulary:	Phrasal Verbs
Listening:	Listening for Main Idea and Supporting Ideas
Speaking:	Getting Someone's Attention and Interrupting
Writing:	Formal Letters

Learning Outcomes:

At the end of the unit, the student will be able to

1. Relate to the world through adult & child's points of view (L3)
2. Identify the sense of self dignity & rationality in the poem (L4)
3. Utilize appropriate phrasal verbs for effective communication(L3)
4. Listen & identify main and supporting ideas in the audio(L3)
5. Practice conversational etiquette(L3)
6. Write formal letters(L3)

UNIT-III **10 Lectures**

Prose:	O. Henry: The Last Leaf
Poetry:	Rudyard Kipling: If
Grammar & Vocabulary:	Idiomatic Expressions
Listening:	Listening for Global Comprehension
Speaking:	Asking for Information and Giving Information
Writing:	Note-Making

Learning Outcomes:

At the end of the unit, the student will be able to:

1. Analyze strategies to face challenges in life (L4)
2. Develop one's own personality (L3)
3. Use idiomatic expressions in oral & written communication (L3)
4. Listen & Interpret the audio for Global Comprehension (L3)
5. Ask & provide information (L3)
6. Practice note making study skills(L3)

UNIT-IV **10 Lectures**

Prose:	Francis Bacon: Of Studies
Poetry:	Toru Dutt: Our Casuarina Tree
Grammar & Vocabulary:	Remedial Grammar I
Listening:	Listening to Make Inferences
Speaking:	Expressing Opinions, and Agreeing and Disagreeing with Opinions
Writing:	Essay Writing

Learning Outcomes:

At the end of the unit, the student will be able to

1. Prioritize the habits of continuous learning(L4)
2. Recognize the significance of Indian philosophy (L4)
3. Identify and correct common errors in English grammar and usage(L3)
4. Draw inferences from the audio(L3)
5. Articulate one's own opinions(L3)
6. Develop different types of essays (DEAN)(L3)

UNIT-V **8 Lectures**

Prose:	Mark Twain: Whitewashing the Fence
Poetry:	William Ernest Henley: Invictus
Grammar & Vocabulary:	Remedial Grammar II
Listening:	Listening for Key Ideas
Speaking:	Telephone Etiquette
Writing:	E-mail Etiquette

Learning Outcomes:

At the end of the unit, the student will be able to

1. Discover the humour & moral lessons in the text (L3)
2. Show resilience in adverse situations. (L3)
3. Modify sentences with appropriate grammar, vocabulary and usage (L3)
4. Identify key ideas (L3)
5. Practice telephone Etiquette (L3)
6. Apply the knowledge of E-mail Etiquette (L3)

Text Books:

1. English for Engineers: Theory to practice. Board of Editors, Orient Black Swan Publishers, India.2024.

Reference Books:

1. English Grammar in Use by Raymond Murphy
2. Oxford English Grammar Course by Michael Swan
3. Word Power Made Easy by Norman Lewis
4. Cambridge Vocabulary for IELTS by Pauline Cullen
5. The Elements of Style by William Strunk Jr. and E.B. White
6. English Vocabulary in Use by Michael McCarthy and Felicity O'Dell
7. Practical English Usage by Michael Swan
8. The Only Grammar Book You'll Ever Need by Susan Thurman
9. Advanced English Grammar: A Linguistic Approach by Ilse Depraetere and Chad Langford

FUNDAMENTALS OF COMPUTERS

I Year B. Tech. I semester
[Common to CSE, CSE (AI&ML), IT]

L	T	P	C
3	0	0	3

Course Code: 24CT11RC01

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

CO1: Identify different I/O devices and their usage. (L3)

CO2: Compare different computer memory with their efficiency. (L3)

CO3: Relate different computer software's and programming languages for solving problems. (L2)

CO4: Summarize the use of operating system, database system, and computer network in the field of computing in engineering. (L2)

CO5: Infer the methods and applications of artificial intelligence (AI), machine learning (ML) and data science. (L3)

UNIT-I **10 Lectures**

Introduction to Computers: Computers, Stored Program Concept, Classification of Computers, Evolution and Development of Computers, Application Areas of Computers, Features/Characteristics of Computers, Basic Organization of a Computer.

Input and Output Devices: Input Devices, Output Devices, Soft Copy Devices, Hard Copy Devices, Voice Response Systems, Biometric Devices [TextBook-1: Chapter 1 (S-1.1 – S-1.7)]

https://oer-studentresources.gesci.org/wp-content/courses/Computer/CS-F1-Introduction-to-computers/evolution_and_development_of_computers.html (Development of Computers) [TextBook-1: Chapter 2 (S-2.1 – S-2.6)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Outline characteristics and classification of computers (L2)
2. Demonstrate different configurations of software and hardware used in computer (L2)
3. Model the computer by organizing different computer peripherals such as I/O, CPU, etc. (L3)

UNIT-II **10 Lectures**

Computer Memory: Introduction, Memory Organization, Memory Hierarchy, Sequential and Random Access, Processor Registers, Cache Memory, Primary Memory, Secondary Storage Devices, Magnetic Tapes, Floppy Disks, Hard Disks, External Hard Disks, Optical Disks, USB Flash Drives, Memory Cards, Mass Storage Devices. [TextBook-1: Chapter 3 (S-3.1 – S-3.16)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Summarize computer memory hierarchy. (L2)
2. Classify different memory with respect to speed and cost. (L2)
3. Illustrate different external storage devices and its applications. (L2)

UNIT-III **10 Lectures**

Computer Software: Introduction to Computer Software, Classification of Computer Software, System Software, Application software, Graphics Software, Multimedia Software, Database Management Software, Firmware, Middleware, Procuring Computer Software.

Problem Solving using Computers: Introduction, Program Development Life Cycle, Program Design Tools, Programming Languages, Generations of Programming Languages, Categorization of High-level Languages, Some Popular High-level Languages, Apps. [TextBook-1: Chapter 6 (S-6.1 – S-6.7), TextBook-1: Chapter 7 (S-7.1 – S-7.7)]

https://www.builder.ai/glossary/app#:~:text=An%20app%20is%20a%20type,enhance%20productivity%20and%20streamline%20communication (Apps)

Learning Outcomes:

At the end of the unit, the student will be able to

1. Compare different computer software's (L2)
2. Summarize different programming languages for solving problems. (L2)
3. Contrast Structured Programming Language, Logic-oriented Programming Language, and Object-oriented Programming. (L2)

UNIT-IV **10 Lectures**

Fundamental Operating Systems: Introduction, History of Operating Systems, Functions of Operating Systems, Types of Operating Systems, Providing User Interface, Popular Operating Systems

Database Systems: File-Oriented Approach, Database-oriented Approach, Database Views, Three-schema Architecture, Database Models, Components of Database Management, Retrieving Data Through Queries.

Computer Networks: Introduction to Computer Networks, Types of Networks, Physical Components of a Network, Wired Media, Wireless Media, Networking Devices, Network Topologies, Wireless Networks. [TextBook-2: Chapter 11 (S-11.1 / P-238 – S-11.11 / P-258), TextBook-1: Chapter 9 (S-9.1 – S-9.7), Chapter 10 (S-10.1 – S-10.8)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Explain fundamentals of operating systems. (L2)
2. Illustrate database system applications and its design concepts. (L2)

3. Outline fundamentals of computer networks and its applications. (L2)

UNIT-V **8 Lectures**

Introduction to Artificial Intelligence & Machine Learning (AI&ML) and Data Science: Introduction to AI & ML, and Data Science, Use Cases in Business and Scope, Modelling Concepts, CRISP-DM Method.

1. https://community.aws/content/2drbbXokwrIXivItJ8ZeCk3gT5F/introduction-to-artificial-intelligence-and-machine-learning?lang=en (Artificial Intelligence)
2. https://www.javatpoint.com/machine-learning-models (Machine Learning)
3. https://www.techtarget.com/searchenterpriseai/feature/Top-12-machine-learning-use-cases-and-business-applications (Machine Learning)
4. https://www.heavy.ai/learn/data-science (Data Science)
5. https://www.sv-europe.com/crisp-dm-methodology/ (CRISP-DM Method)

Learning Outcomes:

At the end of the unit, the student will be able to

1. Outline the main methods and applications of artificial intelligence (AI) concepts. (L2)
2. Outline the main methods and applications of machine learning (ML) concepts. (L2)
3. Outline the main methods and applications of data science concepts. (L2)

Textbooks:

1. Reema Thareja, Fundamentals of Computers, Second Edition, Oxford University Press, 2019.
2. E Balagurusamy, Fundamentals of Computers, Tata McGraw Hill Education Private Limited, 2009

Reference Books:

1. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 4th Edition, McGraw-Hill, 2002.
2. Andrew S. Tanenbaum, Modern Operating Systems, 2nd edition, PHI, 1995.

Web References:

1. https://oer-studentresources.gesci.org/wp-content/courses/Computer/CS-F1-Introduction-to-computers/evolution_and_development_of_computers.html (Development of Computers)
2. https://www.builder.ai/glossary/app#:~:text=An%20app%20is%20a%20type,enhance%20productivity%20and%20streamline%20communication (Apps)
3. https://www.geeksforgeeks.org/what-is-an-operating-system/ (Operating System)
4. https://www.javatpoint.com/computer-network-tutorial (Computer Network)
5. https://community.aws/content/2drbbXokwrIXivItJ8ZeCk3gT5F/introduction-to-artificial-intelligence-and-machine-learning?lang=en (Artificial Intelligence)
6. https://www.javatpoint.com/machine-learning-models (Machine Learning)

7. https://www.techtarget.com/searchenterpriseai/feature/Top-12-machine-learning-use-cases-and-business-applications (Machine Learning)
8. https://www.heavy.ai/learn/data-science (Data Science)
9. https://www.sv-europe.com/crisp-dm-methodology/ (CRISP-DM Method)

PROBLEM SOLVING USING C

I Year B. Tech. I semester

[Common to CSE, CSE (AI&ML), IT, ECE]

L	T	P	C
3	0	0	3

Course Code: 24CT11RC02

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

CO1: Explain the basic constructs of C through the use of simple applications. (L2)

CO2: Demonstrate the utilization of arrays and strings in the development of C programs. (L2)

CO3: Utilize functions and pointers to construct various applications in C. (L3)

CO4: Apply concepts of structures and unions to build and implement C applications. (L3)

CO5: Develop applications using sequential and random-access file processing techniques. (L3)

UNIT-I **10 Lectures**

Introduction to C: Basic structure of C program, Constants, Variables and data types, Operators and Expressions, Arithmetic Precedence and associativity, Type Conversions. Managing Input and Output Operations Formatted Input, Formatted Output. [TextBook-1: Chapter 2 (S-2.1 – S-2.12)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Outline the Basic Structure and Components of C Programs. (L2)
2. Demonstrate Variables, Data Types, Operators, and Expressions. (L2)
3. Illustrate Input and Output Operations. (L2)

UNIT-II **11 Lectures**

Decision Making, Branching, Looping: Decision making with if statement, Simple if statement, The if…else statement, Nesting of if…else statement, the else.. if ladder, switch statement, the (? :) operator, the goto statement, The while statement, the do statement, the for statement, Jumps in Loops.

Arrays & Strings: One, Two-dimensional Arrays, multi-dimensional Arrays, Character Arrays. Declaration and Initialization of Strings, reading and writing of strings, string handling functions. [TextBook-1: Chapter 3 (S-3.1 – S-3.8), TextBook-2: Chapter 13,14,15,16 (PP:215-283)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Classify Decision Making and Branching (L2)
2. Interpretation of Looping Constructs (L2)
3. Summarize Arrays and Strings (L2)

UNIT-III **10 Lectures**

Function and Dynamic Memory Allocation: Functions: Designing structured programs, declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, passing arrays to functions, recursion.

Introduction to Pointers, Pointer Arithmetic, Pointers for Inter-Function Communication, passing pointers to functions, call by reference, Dynamic Memory Allocation. [TextBook-1: Chapter 4 (S-4.1 – S-4.11), Chapter 5 (S-5.1 – S-5.12)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Utilize Functions (Understand and Implement Functions, Utilize Various Function Types, Advanced Function Techniques) (L3)
2. Experiment with Pointers (Basic Pointer Operations, Pointers and Data Structures) (L3)
3. Explain Advanced Pointer Usage (Pointers in Functions, Pointers to Complex Data Types) (L2)

UNIT-IV **9 Lectures**

Structure and Unions: Defining a structure and union, declaring structure variables, accessing structure members, structure initialization, copying and comparing structure variables, arrays of structures, arrays within structures, structures within structures, definition and usage of union, structures and unions using functions, size of structures and bit-fields. The Type Definition (Type def), Enumerated Types. [TextBook-1: Chapter 6 (S-6.1 – S-6.9)]

Learning Outcomes:

1. At the end of the unit, the student will be able to
2. Define Structures and Build C programs using Structures. (L3)
3. Explain Complex Data Structures. (L2)
4. Explain Unions and Bit-Fields. (L2)

UNIT-V **8 Lectures**

File handling: Introduction to Files, Modes of File operations, Text and Binary Files, Defining and opening a file, closing a file, Input/ Output operations on files, Error handling during I/O operations, random access to files and Command Line Arguments, user defined header files. [TextBook-2: Chapter 19 (PP:325-348)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Outline File Operations. (L2)
2. Summarize Error Handling and Random Access. (L2)
3. Utilize Command Line Arguments. (L3)

Text Books:

1. Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", 2nd Edition, 2015, Pearson Education India, ISBN: 978-93-3254-944-9.
2. Yashavant P. Kanetkar, "Let Us C", 16th Edition, 2019, BPB Publications, ISBN: 978-93-8728-449-4.

Reference Books:

1. N. B. Venkateswarlu, E. V. Prasad, "C and Data Structures", 1st Edition, S. Chand Publishing, 2010, ISBN: 978-93-525-3356-5.
2. Pradip Dey, Manas Ghosh, "Programming in C", 2nd Edition, 2018, Oxford University Press, ISBN: 978-01-9949-147-6.
3. Jacqueline A. Jones and Keith Harrow, "Problem Solving with C", Pearson Education. ISBN: 978-93-325-3800-9.
4. E. Balagurusamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316-513-0.

Web References:

1. https://nptel.ac.in/courses/106105171
2. https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/pages/lecture-notes/
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013843237039374336345 17_shared/overview
4. https://cse02-iiith.vlabs.ac.in/List%20of%20experiments.html

COMMUNICATION SKILLS LAB

I Year B. Tech. I semester

[Common to CSE, IT]

Course Code: 24HE11RC02

L	T	P	C
0	0	3	1.5

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

CO1: Identify and pronounce the sounds of English; choose the accurate stress in connected speech for proper intonation. (L3)

CO2: Apply the main theme and ideas of the audio/video to take notes and summarize. (L3)

CO3: Develop speaking skills by taking part in Just A Minute (JAM) – Picture Prompts-Narratives-Role Play. (L3)

CO4: Practice discussions and debates. (L3)

CO5: Demonstrate Presentation Skills. (L3)

ACTIVITY-I

Introduction to Phonetics: The Sounds of English (Speech sound – vowels and consonants) - Stress and Intonation - Accent and Rhythm.

Learning Outcomes:

At the end of the unit, the student will be able to

1. Apply the knowledge of Phonetics for better pronunciation and articulation. (L3)
2. Choose appropriate stress, intonation and rhythm of English language for clear communication. (L3)

ACTIVITY-II

Listening Skills: Listening for gist and specific information - listening for Note taking, Summarizing and for opinions - Listening to the speeches of eminent personalities.

Learning Outcomes:

At the end of the unit, the student will be able to

1. Develop effective listening skills for better comprehension of academic lectures and English spoken by native speakers. (L3).
2. Apply effective strategies for good writing. (L3)
3. Demonstrate writing skills in note taking and summarizing. (L3)

ACTIVITY-III

Speaking Skills: Just A Minute (JAM) session –Picture Prompts- Narrating stories and anecdotes-Role Play

Learning Outcomes:

At the end of the unit, the student will be able to

1. Make use of dialogues for different roles. (L3)
2. Develop communication skills in formal and informal situations. (L3)
3. Practice speaking skills through participation in activities such as narrating stories and role plays(L3)

ACTIVITY-IV

Speaking skills: Group Discussions-Arguments-Debates

Learning Outcomes:

At the end of the unit, the student will be able to

1. Organize one's own ideas for various Group-Discussion formats. (L3)
2. Develop ideas and arguments to debate. (L3)

ACTIVITY-V

Presentation skills: Verbal and non-verbal communication - Body Language - Making a Presentation

Learning Outcomes:

At the end of the unit, the student will be able to

1. Design presentations with PowerPoint slides(L3)
2. Apply appropriate body language (postures, gestures, facial expressions and eye contact) in formal presentations. (L3)

LIST OF LAB ACTIVITIES:

1. Identification and pronunciation of Vowel sounds and Consonant sounds (CO1)
2. Identification of word stress, Intonation and Rhythm (CO1)
3. Listening for specific information & Note taking (CO2)
4. Listening to the speeches of eminent personalities and summarizing (CO2)
5. Just A Minute sessions (CO3)
6. Picture Prompts (CO3)
7. Narrating Stories& anecdotes (CO3)
8. Role-Plays (CO3)
9. Group Discussions (CO4)

10. Debates (CO4)
11. Presentation Skills-I (CO5)
12. Presentation Skills-II (CO5)

Reference Books:

1. Language and Life: A Skills Approach Board of Editors, Orient Black swan Publishers, India.2018.
2. A Textbook of English Phonetics for Indian Students, T.Balasubramanian, Macmillan India Ltd.
3. Ashraf Rizvi. Effective Technical Communication. Tata McGraw Hill Education Private Limited, New Delhi.
4. Speak Well. Orient Black swan Publishers, Hyderabad.
5. Allan Pease. Body Language. Manjul Publishing House, New Delhi.

Web References:

1. https://www.englishlanguageclub.co.uk
2. https://www.ted.ed.com/
3. https://learningenglish.voanews.com/
4. https://www.bbc.co.uk/learningenglish/
5. https://www.abc.net.au/education/learn-english
6. NDTV News

COMPUTER ENGINEERING WORKSHOP

I Year B. Tech. I semester
[Common to CSE, CSE (AI&ML), IT]

Course Code: 24CT11RC03

L	T	P	C
0	0	3	1.5

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

CO1: Identify the peripherals of a computer. (L3)

CO2: Demonstrate the installation process of operating systems like MS Windows, LINUX. (L3)

CO3: Make use of MS-Office suite that comprises of various applications to create documents, organizes & analyses data and captivating presentations. (L3)

CO4: Categorize various UNIX/LINUX Commands to work on a modern operating system. (L3)

CO5: Experiment with GIT Commands and make use of LaTeX to prepare research articles. (L3)

Module-1: Hardware Concepts

1. Identify the peripherals of a computer, components in a CPU and its functions.
2. Draw the block diagram of the CPU along with the configuration of each peripheral.
3. Disassemble and assemble the PC back to working condition.

Module-2: Software Installations

1. Install operating systems like LINUX and MS windows on a personal computer.
2. The system should be configured as dual boot with both MS windows and LINUX.

Module-3: Office Suite

1. Creating Documents: Formatting Fonts/Texts, page layout, Applying Text effects, Character Spacing, Borders and Colours, Inserting Header and Footer, equation editor.
2. Spreadsheet Orientation: Accessing, overview of toolbars, saving spreadsheet files, Using help and resources. Creating a Scheduler, Gridlines, Format Cells, Summation, auto fill, Formatting Text, Renaming and Inserting worksheets, Hyper linking.
3. Excel Calculation: Calculating Average, Cell Referencing, Excel Formula, std. deviation, Charts, Count Function, Sorting, Conditional formatting, Excel Advanced-sumif, VLOOKUP, and Pivot Table.
4. Creating Power Point: Student should work on basic power point utilities and tools to create basic power point presentation. PPT Orientation, Slide Layouts, Inserting Text, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows, Hyperlinks, Inserting Images, Tables and charts.
5. Software as a service (SaaS): Usage of online office suite.

Module-4: Unix / Linux Commands

1. Study and practice on file system, handling files with commands, syntax, usage, application.
2. Practice on vi editor.
3. Study and practice on shells/shell programming with relevant programming constructs, syntax, usage, application.

Module-5: GIT Commands and LaTeX

1. Installation of GIT, Branch in Git Basic GIT Commands.
2. Installation of LaTeX and preparing research articles by creating overleaf account.
3. Introduction to Scratch Programming - Introduction, Stage, Sprite, Script, Block Palette.

Reference Books:

1. IT Essentials: PC Hardware and Software Companion Guide, Third Edition, David Anfinson, Ken Quamme.
2. UNIX concepts and applications by Sumitabha Das, TMH Publications.
3. Shell programming by Yashwanth Kanetkar.

Web References:

1. https://www.javatpoint.com/

PROBLEM SOLVING USING C LAB

I Year B. Tech. I semester

[Common to CSE, CSE (AI&ML), IT, ECE]

Course Code: 24CT11RC04

L	T	P	C
0	0	3	1.5

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

CO1: Outline the use of basic constructs of C for simple applications. (L2)

CO2: Develop C programs for simple applications using Arrays and Strings. (L3)

CO3: Illustrate concepts such as functions, recursion, and pointers with suitable examples. (L2)

CO4: Build C programs involving Structures and Unions. (L3)

CO5: Develop applications using sequential and random-access file processing. (L3)

Module-1:

1. Write a C program to demonstrate Format Specifiers and Input/Output Statements.
2. Write a C program to demonstrate various Data Types.

Module-2:

1. Write a C program to demonstrate various Operators including Bitwise Operator.
2. Write a C program to demonstrate Arithmetic Expressions and Type Casting.

Module-3:

1. Write a C program to demonstrate decision making statements.
2. Write a C program to demonstrate iterative statements.
3. Write a C program to demonstrate break and continue Statement.

Module-4:

1. Write a C program to demonstrate arrays (one-dimensional and two-dimensional).
2. Write a C program to demonstrate string handling functions using built-in and user defined functions.

Module-5:

1. Write a C program to demonstrate user defined functions.
2. Write a C program to demonstrate nested functions.

Module-6:

1. Write a C program to demonstrate arithmetic expressions using pointers.
2. Write a C program to demonstrate Pointers to Arrays.
3. Write a C program to demonstrate manipulate strings using pointers.
4. Write a C program to demonstrate dynamic memory allocation.

Module-7:

1. Write a C program to demonstrate Call-by-value, Call-by-reference.
2. Write a C program to demonstrate recursive function.
3. Write a C program to demonstrate Functions with Storage Classes (Static).

Module-8:

1. Write a C program to demonstrate structures.
2. Write a C program to demonstrate structures within structures.
3. Write a C program to demonstrate pointer to a structure.
4. Write a C program to demonstrate unions.

Module-9:

1. Write a C program to demonstrate I/O operations on files.
2. Write a C program to demonstrate concatenating two files.
3. Write a C program to demonstrate copy content of one file to another file.

Case Study: Select any one application mentioned below.

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

1. Develop a library management system to add, delete, and search for books efficiently.
2. Develop an ATM system to check balance, deposit money, and withdraw funds.
3. Develop a C program to calculate salary increments using a lookup table which stores the percentage of increment based on the current salary.
4. Develop a voting system where users can vote for candidates and view results.
5. Develop a basic calculator for arithmetic operations like addition, subtraction, multiplication, and division.
6. Develop a contact management system to store and manage details like name, phone number, and email.
7. Develop a banking system supporting multiple users for account management and transactions.
8. Develop a C program to calculate SGPA and CGPA based on students' marks across semesters.
9. Develop a registration page in C using file handling to securely store usernames and passwords.
10. Develop an encryption and decryption algorithm to securely transform and restore data by using the GNU C Library: crypt.

Reference Books:

1. Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", 2nd Edition, 2015, Pearson Education India, ISBN: 978-93-3254-944-9.
2. Yashavant P. Kanetkar, "Let Us C", 16th Edition, 2019, BPB Publications, ISBN: 978- 93- 8728-449-4.
3. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316-513-0.
4. Pradip Dey, Manas Ghosh, "Programming in C", 2nd Edition, 2018, Oxford University Press, ISBN: 978-01-9949-147-6.
5. N.B. Venkateswarulu, "C programming", 1st Edition, S.Chand Publishing, 2017, ISBN: 978-93-525-3356-5.
6. Jacqueline A Jones and Keith Harrow, "Problem Solving with C", Pearson Education. ISBN: 978-93-325-3800-9.

Web References:

1. https://nptel.ac.in/courses/106105171
2. https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/pages/lecture-notes/
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013843237039374336345 17_shared/overview
4. https://cse02-iiith.vlabs.ac.in/List%20of%20experiments.html

LINEAR ALGEBRA AND VECTOR CALCULUS

I Year B. Tech. II Semester

[Common to EEE, ECE, CSE, IT, CSE (AI&ML)]

Course Code: 24BM11RC02

L	T	P	C
3	0	0	3

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

CO1: Test for consistency and solve linear system of equations, also determine unknown currents in the electrical circuits. (L5)

CO2: Find the Eigen values and Eigen vectors of a matrix and apply Cayley- Hamilton theorem to find the inverse of a matrix. (L3)

CO3: Reduce quadratic form to canonical form and examine the nature of quadratic form. (L4)

CO4: Interpret the meaning and evaluate gradient of a scalar valued, curl and divergence of vector valued functions. (L5)

CO5: Apply line integrals, surface integrals, volume integrals and their relations using Green's theorem, Stoke's theorem, Gauss Divergence theorems in various engineering applications. (L3)

UNIT-I: **10 Lectures**

Matrix Algebra: Rank of a matrix- Echelon form, Normal Form - Solution of Linear System of Equations - Consistency of Linear System of Equations – Gauss elimination and Gauss Jordan methods, LU Factorization method.

Applications: Finding the current in electrical circuits. [Sections: 2.7, 2.10, 28.6 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Find the rank of a matrix. (L1)
2. Test for consistency and solve a system of linear equations. (L4)
3. Apply Gauss elimination and Gauss Jordan methods, LU factorization to solve linear systems. (L3)
4. Determine unknown currents in electrical circuits. (L5)

UNIT-II: **10 Lectures**

Eigen Values and Eigen Vectors: Eigen Values and Eigen Vectors of a real Matrix – Properties - Cayley- Hamilton theorem (without proof) - Inverse and Powers of a Matrix using Cayley-Hamilton's theorem – Pseudo inverse of a matrix, Singular value decomposition. [Sections: 2.13 - 2.15 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Find eigenvalues and corresponding eigenvectors of a matrix. (L1)
2. Apply Cayley-Hamilton theorem to find powers and the inverse of a matrix. (L3)
3. Calculate the Pseudo Inverse of a matrix. (L3)
4. Calculate the singular values of a matrix. (L3)

UNIT-III: **9 Lectures**

Quadratic Forms: Inner Product – Orthogonal Vectors – Orthogonal matrix – Diagonalization of a Matrix-Quadratic Forms - Reduction of Quadratic Form to Canonical Form (Orthogonal Transformation) - Nature of a Quadratic Form. [Sections: 2.16 - 2.18 of Textbook]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Make use of Eigen values and eigen vectors to diagonalize the matrix. (L3)
2. Reduce the Quadratic form to canonical form examine the nature of a quadratic form. (L4)

UNIT-IV: Vector Differentiation **9 Lectures**

Introduction - Scalar and Vector point functions, General rules for vector differentiation - Vector operator ∇ applied to scalar point functions- Gradient, ∇ applied to vector point functions- divergence and curl. Physical interpretation of gradient, divergence and curl(i.e., ∇f, $\nabla . \overline{F}$, $\nabla \times \overline{F}$), Irrotational and Solenoidal fields, Vector Identities. [Sections: 8.1, 8.4 - 8.9 of Text Book]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Find the gradient of a scalar point function, divergence and curl of a vector point function. (L1)
2. Determine the directional derivative of scalar point function. (L5)

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

Integration of vectors - Line integral – Circulation - work done - surface integral-flux - Green's theorem in the plane - Stoke's theorem - Volume integral - Gauss Divergence theorem (All theoremswithout proofs). [Sections: 8.10 - 8.16, 8.18 of Text Book]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Determine the work done in a moving particle along a path. (L5)
2. Interpret surface and volume integrals. (L2)

3. Apply vector integral theorems to multiple integral. (L3)

Textbooks:

1. B. S. Grewal, Higher Engineering Mathematics, 45th Edition, Khanna Publishers, 2024.

Reference Books:

1. David Poole, Linear Algebra- A modern introduction, 4th edition, 2015.
2. Erwin Kreyszig, Advanced Engineering Mathematics, 10th edition, John Wiley & Sons, 2011.
3. Peter V. O'Neil, Advanced Engineering Mathematics, 7th edition, Cengage Learning, 2011.
4. Greenberg, Advanced Engineering Mathematics, 2nd Edition, Pearson, 2017.

Web References:

1. https://nptel.ac.in/courses/111107112

ENGINEERING PHYSICS

I Year B. Tech. II Semester

[Common to CSE, IT]

L	T	P	C
3	0	0	3

Course Code: 24BP11RC01

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

CO1: Apply the principles of interference, diffraction, and polarization in wave optics, and utilize optical instruments for practical applications. (L3)

CO2: Familiarize the basic concepts of Thermodynamics relevant to engineering applications. (L2)

CO3: Outline the knowledge of basic principles of Electromagnetism and EM Waves. (L2)

CO4: Describe the basic principles of lasers, optical fibres and their use in technological applications. (L2)

CO5: Understand the basic principles of Quantum mechanics, Quantum computing. (L2)

UNIT-I **10 Lectures**

Interference: Principle of superposition, Young's Experiment (qualitative treatment), Coherence, Interference in thin films (reflected light), Newton's Rings, Michelson's Interferometer and its applications (thickness of thin sheet, determination of wavelength).

Diffraction: Introduction, Differences between Interference and Diffraction, Differences between Fresnel and Fraunhoffer diffraction, Fraunhoffer diffraction at a single slit (Qualitative and Quantitative treatment)

Polarization: Introduction - types of Polarization. Polarization by reflection, Brewster's law, refraction and double refraction in uniaxial crystals, Nicol Prism, Quarter wave plate and Half wave plate, Applications of Polarization.

Text Book 1: 5.16,5.17,5.18, 5.20, 6.1, 6.2, 6.8.(1-3), 6.11, 6.13, 6.14.1, 6.14.3, 7.1, 7.2, 7.3, 7.4, 7.5, 8.1, 8.2,8.5, 8.6,8.11,8.12, 8.16,8.20.

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand the principles of interference patterns and apply Michelson's Interferometer to measure physical quantities such as thickness and wavelength. (L2)
2. Distinguish between interference and diffraction and perform qualitative and quantitative analysis of Fraunhoffer diffraction at a single slit. (L2)
3. Apply polarization concepts through various optical devices to explore and solve problems related to polarized light. (L3)

UNIT-II **7 Lectures**

Thermodynamics: Heat and Work, First law of thermodynamics and its applications, Reversible and Irreversible process, Carnot cycle and Efficiency, Second law of thermodynamics, Carnot's Theorem, Entropy, Second law in terms of Entropy, Entropy and disorder, Third law of thermodynamics (statement only). [TextBook-1: 16.3,16.5.1,16.6,16.8,16.11,16.12,16.14,16.16,16.17, 16.17.1, 16.18]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Apply the first and second laws of thermodynamics to analyse energy transformations and process efficiencies. (L3)
2. Understand the Carnot cycle and its efficiency and calculate entropy changes to explain thermodynamic processes. (L2)
3. Differentiate between reversible and irreversible processes and relate entropy to disorder. (L2)
4. State the third law of thermodynamics and understand its implications for materials at absolute zero. (L2)

UNIT-III **11 Lectures**

Electromagnetism: Concept of electric flux, Gauss' law, applications of Gauss' law (wire, sheet, sphere), Magnetic Field, Gauss' law in magnetostatics, Faraday's law of induction, Lenz's law, Induced magnetic fields, Displacement current, Maxwell's equations (no derivation, qualitative treatment), Electromagnetic wave equations, Introduction to EM waves, The Biot-Savart's law, magnetic field near a long wire, magnetic field for a circular Current loop, Ampere's law. [TextBook-1: 2.12,2.14,2.18, 3.5, 3.8, 3.9,3.12, 3.13,3.14, 3.10, 3.12, 3.3,3.4]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Apply Gauss's law and Faraday's law of induction to solve problems involving electric and magnetic fields. (L3)
2. Use Maxwell's equations to understand electromagnetic wave phenomena. (L2)
3. Analyse magnetic fields using Biot-Savart's law, Ampere's law, and apply e magnetostatic field distributions. (L3)

UNIT-IV **10 Lectures**

Lasers: Characteristics of laser beam, Spontaneous and stimulated emission of radiation, population inversion, Ruby laser, He-Ne laser, Semiconductor laser (homojunction), Applications of lasers.

Optical fibres: principle of propagation of light in optical fibres, Acceptance Angle and cone of a fibre, Numerical aperture, Modes of propagation, Classification of fibres based on refractive index profile and modes, Losses in optical fibres- scattering and bending losses, Fibre optics in communication- Block diagram, Applications of optical fibres. [TextBook-1: 24.12, 24.2, 24.5,24.6, 24.7, 24.11.1, 24.11.3, 24.11.5.1, 24.13, 10.2, 10.3, 10.4, 10.5, 10.6, 10.10, 10.11, 10.19,10.20 (only for list of applications]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand the principles of laser operation, including spontaneous and stimulated emission, and analyse various types of lasers such as Ruby, He-Ne, and Semiconductor lasers. (L2)
2. Describe the fundamentals of optical fibre technology, including light propagation, acceptance angle, numerical aperture, and classification of fibres. (L2)
3. Apply these concepts to fibre optics communication systems and explore their practical applications. (L2)

UNIT-V **10 Lectures**

Quantum Physics: de Broglie concept of matter waves, Heisenberg's uncertainty principle, Schrödinger time independent wave equation, Physical significance of wave function, particle in a one-dimensional box.

Quantum Computing Quantum bits, Introduction to Pauli spin matrices, Bloch sphere, Entanglement, Qubit Vs classical bit, Single Qubit gates, Quantum Teleportation – Basic Idea. [Textbook-1 20.5, 20.11, 20.17, 20.18, 20.22; Textbook -3 Sec 1.8, 1.9, 1.10.1]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand the concepts quantum mechanics such as the de Broglie wavelength and Heisenberg's uncertainty principle and use the Schrödinger equation (L2).
2. Describe the fundamentals of quantum computing (L2)

Textbooks:

1. M. N. Avadhanulu, P. G. Kshirsagar, and T. V. S. Arun Murthy, A textbook of Engineering Physics, 11th edition, S. Chand and Company Ltd., 2019.
2. H. K. Malik and A. K. Singh, Engineering Physics, 2nd Edition, McGraw Hill Education Pvt Ltd. ,2018.
3. P.K. Palanisamy, Engineering Physics, SCITECH Publications, 2011.

Reference Books:

1. Modern Engineering Physics by A.S. Vasudeva S. Chand and Company Ltd., 2010
2. University Physics by Young and Freedman Pearson Education, 2010.
3. Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information Cambridge University Press, 10 th Ed., 2010

Web References:

1. https://nptel.ac.in/courses/115105537 (Wave optics)
2. https://nptel.ac.in/courses/115106122 (Electromagnetism)
3. https://nptel.ac.in/courses/112102255 (Thermodynamics)
4. https://nptel.ac.in/courses/115107095 (Fiber Optics)
5. https://nptel.ac.in/courses/104104085 (Lasers)
6. https://nptel.ac.in/courses/104104082 (Quantum Computing)
7. https://www.intechopen.com/online-first/73811 (Quantum Computing)
8. https://lewisla.gitbook.io/learning-quantum/quantum-circuits/single-qubit-gates(Quantum Computing)
9. https://profmcruz.wordpress.com/wp-content/uploads/2017/08/quantum-computation-and-quantum-information-nielsen-chuang.pdf (Quantum Computing)

DIGITAL LOGIC DESIGN

I Year B. Tech. II semester

[Common to ECE, CSE, CSE(AI&ML), IT]

Course Code: 24EC11RC05

L	T	P	C
3	0	0	3

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

CO1: Represent different number systems & binary codes and perform conversions & binary arithmetic. (L2)

CO2: Apply different simplification methods for minimizing Boolean functions. (L3)

CO3: Model various combinational circuits using gates and PLD's. (L3)

CO4: Outline the concept of latches and flip-flops. Construct sequential logic circuits like counters and registers using flip-flops. (L3)

CO5: Categorize Mealy & Moore models and Design Synchronous Sequential machines. (L3)

UNIT-I **9 Lectures**

Number systems: Number systems, Base conversion methods, Representation of signed numbers and Binary Arithmetic.

Codes: Binary, Non binary, Decimal, Alphanumeric, XS-3, Gray. Error detecting and error correcting codes.

Logic Gates: AND, OR, NOT, NAND, NOR, XOR, EX-NOR. [Textbook1: Chapter1, Chapter 2: section 2.1 to 2.8]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand the advantages of using different number systems. (L2)
2. Describe the usefulness of different binary codes. (L2)
3. Summarize the error detection and correction concepts. (L2).

UNIT-II **10 Lectures**

Minimization of Boolean Functions: Fundamental postulates of Boolean algebra, Basic theorems, Simplification of Boolean equations, Min terms, Max terms, Standard form of Boolean functions. Simplification of functions using Karnaugh map method (2,3,4,5 variables) Don't care conditions, AOI implementation, NAND and NOR Realizations. [Textbook1: Chapter 2: Sections 2.11 to 2.19,2.21,2.22,2.23, Chapter 3: sections 3.2 to 3.6,3.7]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Apply basic laws and theorems to simplify Boolean expressions and construct logic circuits. (L3)
2. Understand concepts of sum-of-products and product-of-sums representations. (L2)
3. Apply K- Map for minimizing logic functions and build logic circuits. (L3)

UNIT-III **12 Lectures**

Combinational Logic-Circuit Design-I: Logic design of combinational circuits: Adders and Subtractors: Binary, BCD, Excess-3 and Look–ahead-carry adder, Code converters, Multiplexers, De multiplexers, Encoders, Decoders and priority encoders, Realization of Boolean functions using multiplexers and Decoders.

Combinational Logic-Circuit Design-II: Design of 4-bit comparator, Parity checker/Generator. Basics of PLDs: Basic structure of PROM, PAL, PLA, Realization of Boolean functions with PLDs and their merits and demerits. [Textbook1: Chapter 4: section 4.1 to 4.27, Chapter 5 Sections 5.1 to 5.11]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Apply Boolean algebra for describing combinational digital circuits (L3)
2. Describe standard combinational circuits such as adders, subtractors, comparators etc. (L2)
3. Develop the digital circuits using PLDs (L3)

UNIT-IV **10 Lectures**

Sequential Circuits: Classification of sequential circuits, SR-latch, Gated latches, Flip flops: RS, JK, D, T and Master slave flip flops, Excitation tables, flip flop conversion from one type to another. Design of counters: Ripple counters, Synchronous counters, asynchronous counters, up-down counters, Johnson counter, ring counter. Design of registers: Buffer registers, Shift registers, Bi directional shift registers, Universal shift register. [Textbook1: Chapter 6]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand the principle of Flip-Flops and Latches. (L2)
2. Summarize the concepts of Shift Registers and Counters. (L2)
3. Construct different sequential logic circuits using Flipflops. (L3)

UNIT-V **7 Lectures**

Analysis and Design of Synchronous Sequential Machines: Moore and Mealy machine models, State Equations, State Table, State diagram, State reduction & assignment, Synthesis of synchronous sequential circuits- sequence detector and generator. [Textbook1: Chapter 7 Sections 7.1 to 7.5]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand Moore and Mealy machine models (L2)
2. Discuss the concepts of State assignment & Reduction (L2)
3. Analyse the design and synthesis of synchronous sequential circuits (L3)

Textbooks:

1. Anand Kumar, Switching Theory and Logic Design. PHI, 2014.
2. M. Morris Mano and Michael D. Ciletti, Digital Design, 4th Edition, Pearson Education, 2013.

Reference Books:

1. Foundation of Switching theory and Logic Design, A k Singh, New age International Publishers,2008
2. Modern Digital Electronics, R P Jain, 4th Edition, Tata McGraw Hill Education Pvt. Ltd, New Delhi,2010
3. Fundamentals of Logic Design, Charles. R. Roth, Thomson Publications,5th edition,2004.

Web References:

1. https://nptel.ac.in/courses/108105132 (Digital Electronic Circuits)
2. https://onlinecourses.nptel.ac.in/noc24_ee147/preview (Digital Circuits)
3. https://nptel.ac.in/courses/117105080 (Digital Systems Design)
4. https://onlinecourses.swayam2.ac.in/nou24_ec07/preview (Digital electronic and System design)

ELEMENTS OF ELECTRONICS ENGINEERING

I Year B. Tech. II Semester

[Common to CSE, IT]

Course Code: 24EC11RC04

L	T	P	C
3	0	0	3

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

CO1: Outline the concepts of semiconductor devices. (L2)

CO2: Understand the characteristics of different semiconductor diodes and their applications. (L2)

CO3: Apply various biasing techniques for design of BJT circuits. (L3)

CO4: Apply hybrid models to understand the operation of BJT amplifier. (L3)

CO5: Understand the characteristics of FET. (L2)

UNIT-I 8 Lectures

Introduction to Electronics and Semiconductors: Energy band theory, Conduction in Insulators, Semiconductors and metals, Classification of semiconductors, Properties of intrinsic and extrinsic semiconductor, Carrier concentration in an intrinsic semiconductor, Drift, and diffusion currents. [Textbook 1: Chapter -1 Section 1.6, 1.7, 1.8; Chapter -2 Section 2.2, 2.3, 2.9; Chapter -19 Section 19.5]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Explain the energy band theory of solids (insulators, semiconductors, and metals) based on their band structures (L2)
2. Describe the transport phenomena in semiconductors (L2)

UNIT-II 11 Lectures

Semi-Conductor Diode: Theory of PN junction, Open circuited PN junction, V-I characteristics of a PN diode, Diode current equation, basics on junction capacitance, Break down in PN diode, Applications of PN diodes. Zener diode, Zener regulator, Tunnel diode, Schottky diode.

Rectifying circuits: Half wave and full wave rectifiers, Bridge rectifiers, Efficiency, Ripple and regulation of each rectifier, Capacitor filters. [Textbook 2: Chapter 4 Section 4.1 – 4.5, 4.8; Chapter 5 Section 5.3, 5.4, 5.7; Chapter 8 Section 8.2 – 8.4, 8.6]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Explain the formation of PN junction and represent the band structure, interpret the Volt-ampere characteristics (L2)

2. Describe the operation of rectifiers, and evaluate their performance metrics like ripple factor and regulation characteristics (L2)
3. Explain the working principles of special diodes such as Zener diode, tunnel diodes and Schottky barrier diodes. (L2)

UNIT-III **11 Lectures**

Bi-Polar Junction Transistor: Introduction, construction, Operation of PNP and NPN Transistors, transistor configurations, Input and Output Characteristics of CE configuration, Comparison of CE, CB, and CC Configurations.

Transistor Biasing and thermal stabilization: Junction Biasing for Saturation, Cutoff and Active Region, Biasing circuits analysis, fixed bias, collector to base bias, emitter bias, voltage divider bias, thermal runaway. [Textbook 2: Chapter 6 Section 6.1 - 6.3, 6.6-6.9, 6.11, Chapter 10 Section 10.4 - 10.8]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Describe the characteristics of transistor in CE, CB, and CC configurations. (L2)
2. Understand the principals of transistor biasing and compensation methods (L2)
3. Identify the characteristics of different transistor biasing techniques (L3)

UNIT-IV **8 Lectures**

Transistor Amplifiers: CE, CB, CC amplifier configurations –Multistage amplifier – A Two Stage RC coupled amplifier – frequency response curve and bandwidth. [Textbook 2: Chapter -11 Section 11.9, 11.11, 11.12, 11.16, 11.17, 11.20]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Represent transistor with its equivalent h parameter model (L2)
2. Apply hybrid models to obtain the performance of transistor amplifiers (L3)

UNIT-V **10 Lectures**

Field Effect Transistors: Construction of JFET and its drain, transfer characteristics, pinch off Voltage, Drain Saturation Current, parameters of JFET, Construction of MOSFET – Enhancement and Depletion Modes, drain and transfer characteristics. [Textbook 2: Chapter -7 Section 7.1 – 7.5, 7.11, 7.12]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Comprehend the characteristics of BJT and FET(L2)
2. Explain the construction of JFET and MoSFETs (L2)
3. Describe the characteristics of JFET and MoSFET (L2)

Textbooks:

1. Jacob Millman, Christos C. Halkias, Integrated Electronics: Analog and Digital Circuits and Systems, McGRAW-HILL. 1972
2. Sanjeev Gupta, Electronic Devices and Circuits, 2nd Edition, Dhanpat Rai Publications,2005.

Reference Books:

1. Thomas L.Floyd, Electronic Devices, 9th Edition , Prentice Hall, 2012.
2. B. V. Rao and K. Raja Rajeswari, Electronic Devices and Circuits 2nd Edition, Pearson Education.,2010.
3. David A. Bell, Electronic Devices and Circuits, 5th Edition, Oxford University Press, 2008
4. Robert Boylestad, Louis Nashelsky, Electronic Devices and Circuit Theory, 7th Edition, Prentice Hall,1998

Web References:

1. https://onlinecourses.nptel.ac.in/noc21_ee80/preview
2. https://be-iitkgp.vlabs.ac.in/

PYTHON PROGRAMMING

I Year B. Tech. II Semester

[Common to CSE, IT, CSE (AI&ML)]

Course Code: 24CT11RC06

L	T	P	C
3	0	0	3

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

CO1: Understand the historical development, necessity, applications, and basic concepts of python programming. (L2)

CO2: Identify and implement appropriate control structures to solve a particular programming problem. (L3)

CO3: Examine various data structures and apply to solve real world problems. (L3)

CO4: Build simple functions and packages used in python for solving real world problems. (L3)

CO5: Outline Object Oriented concepts in python and illustrate Exception handling. (L2)

UNIT-I **8 Lectures**

Introduction: History of Python, Need of Python Programming, Applications of Python, Variables, Assignment, Comments, Keywords, Data types, Input-Output, Indentation. [TextBook-1: Chapter 3 (S-3.1 – S-3.12)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand the Evolution of Python [L2]
2. Recall the Need and Applications of Python [L1]
3. Outline python's Input and Output Functions[L2]

UNIT-II **10 Lectures**

Operators and Expressions: Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations, Control Flow- if, if-elif-else, for, while, break, continue, pass [TextBook-1: Chapter 3 (S-3.12), Chapter 4 (S-4.1 – S-4.7)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Illustrate python operators: arithmetic, comparison, assignment, logical, bitwise, membership, identity [L2]
2. Make use of control flow statements: if, if-elif-else, for, while, break, continue, pass. [L3]

UNIT-III **10 Lectures**

Data Structures: Strings, Lists, Tuples, Sets, Dictionaries. Data Structures manipulations - create, Index, Negative indexing, Slicing, update, add elements, delete or remove elements, operations, Comprehension, Membership Test, Iteration. [TextBook-1: Chapter 6 (S-6.1 – S-6.9), Chapter 8 (S-8.1 – S-8.6)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understanding basic operations on lists, including indexing and slicing [L2]
2. Utilize methods of lists, tuples, sets, and dictionaries to solve problems [L3]
3. Compare and apply appropriate use cases for lists, tuples, sets, and dictionaries. [L2]

UNIT-IV **10 Lectures**

Functions - Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Lambda Functions, Function Returning Values, Built-in functions, Scope of the Variables in a Function - Global and Local Variables.

Modules: Creating modules, import statement, from…import statement, namespaces. Python packages: Introduction to PIP, Installing Packages via PIP [TextBook-1: Chapter 5 (S-5.1 – S-5.8,5.12)]

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understanding and implementing various types of function arguments and scopes in Python. [L2]
2. Utilize Python modules to understanding the import mechanisms [L3]

UNIT-V **10 Lectures**

Object Oriented Programming (OOP) in Python: Classes and Objects, Class Method and self-Argument, The __init() Method (The Class Constructor), Class Variables and Object Variables, Public and Private Data Members, Inheritance, Overriding Methods.

Error and Exceptions: Introduction to Errors and Exceptions, Handling Exception, try except block, finally block, Raising Exceptions.

File Handling: Type of files, Opening and Closing files, Reading and Writing files.

TextBook-1: Chapter 9 (S-9.1 – S-9.8), Chapter 10 (S-10.1 – S-10.3)

TextBook-1: Chapter 7 (S-7.1 – S-7.5), Chapter 12 (S-12.1 – S-12.10)

Learning Outcomes:

At the end of the unit, the student will be able to

1. Understand and implement classes, methods, and the 'self' variable in Python [L2]
2. Demonstrate concepts of inheritance, method overriding, and data hiding in OOP [L2]

3. List the differences between errors and exceptions, and handle exceptions using try-except blocks [L1]
4. Illustrate file operations including reading, writing, and manipulating file pointers in Python. [L2]

Textbooks:

1. Reema Thareja, “Python Programming: Using Problems Solving Approach”, Oxford University Press, 2017
2. Fundamentals of Python First Programs, Kenneth. A. Lambert, Cengage.

Reference Books:

1. Fundamentals of Data Structures using Python, P Lalitha Surya Kumari, P. S. Avadhani, Lambert Academic Publishing
2. Introduction to Python Programming, Gowrishankar.S, Veena A, CRC Press.
3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

Web References:

1. https://www.tutorialspoint.com/python3/python_tutorial.pdf
2. https://www.python.org/doc/

ENGINEERING PHYSICS LAB

I Year B. Tech. II semester

[Common to CSE, IT]

Course Code: 24BP11RC02

L	T	P	C
0	0	3	1.5

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

CO1: Interpret the physical parameters based on optical phenomena. (L2)

CO2: Verify the behaviour of double refraction in birefringent material like quartz. (L2)

CO3: Demonstrate the concepts of diffraction through experiments. (L3)

CO4: Calibrate instruments like low range voltmeters and ammeters. (L3)

CO5: Design temperature sensors based on diodes and thermistors. (L3)

List of Experiments: (Any TEN of the following experiments shall be conducted)

1. Determination of Radius of Curvature of a given Convex Lens By forming Newton's Rings.
2. Determination of Thickness Given Paper Strip or hairline by Wedge Method.
3. Determination of Wavelength of Spectral Lines in the Mercury Spectrum by Normal Incidence method
4. Determination of Cauchy's Constants of a Given Material of the Prism using Spectrometer in minimum deviation position
5. Determination of Refractive Index of Ordinary ray and Extra-ordinary ray in quartz prism
6. Laser- Diffraction – determination of wavelength of laser source using diffraction grating
7. Determination of wavelength of laser source using Fraunhofer single slit diffraction
8. Melde's Apparatus – Frequency of electrically maintained Tuning Fork.
9. Study of Intensity Variation of the Magnetic Field along the axis of circular Current Carrying conductor using Stewart and Gee apparatus
10. Calibration of Low Range Voltmeter using Potentiometer Bridge
11. Calibration of Low Range Ammeter using Potentiometer Bridge
12. Determination of dielectric constant of material using charging discharging method with a parallel plate capacitor
13. Determination of energy band gap of semiconductor using PN junction diode in reverse bias
14. Determination of thermo electric coefficients for thermistor
15. Determination of Planck's constant using LED.

Textbooks:

1. Practical physics by CL Arora, S.chand publishing company, 1995.
2. Advanced Practical Physics For Students by B.l.worsnop And H.t.flint, 1923

Web References:

1. http://lo-au.vlabs.ac.in/laser-optics/Newtons_Rings_Wavelength_of_light/
2. http://ov-au.vlabs.ac.in/optics/Spectrometer_Refractive_Index/
3. http://ov-au.vlabs.ac.in/optics/Diffraction_Grating/
4. http://htv-au.vlabs.ac.in/heat-thermodynamics/Characteristics_of_Thermistor/
5. https://bop-iitk.vlabs.ac.in/exp/energy-band-gap/simulation.html

PYTHON PROGRAMMING LAB

I Year B. Tech. II semester

[Common to CSE, CSE (AI&ML), IT]

Course Code: 24CT11RC07

L	T	P	C
0	0	3	1.5

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

CO1: Understand core programming basics and various Operators of Python. (L2)

CO2: Implement programs using conditional statements and loops and strings. (L3)

CO3: Develop functions and strings to perform simple tasks. (L3)

CO4: Make use of various data structures like lists, tuples, sets and dictionaries. (L3)

CO5: Implement Python programs with files, Classes and objects. (L3)

Module-1: Basics of Python

1. Write a program to display the statements.
2. Demonstrate about fundamental Data types in Python Programming. (i.e., int, float, complex, bool and string types)
3. Write a Python program to demonstrate various type conversion functions.
4. Write a program to demonstrate the inbuilt Math function

Module-2: Operators in Python

1. Demonstrate the following Operators in Python with suitable examples.

   i) Arithmetic Operators ii) Relational Operators

   iii) Assignment Operators iv) Logical Operators

   v) Bit wise Operators vi) Ternary Operator

   vii) Membership Operators viii) Identity Operators

Module-3: Conditional Branching Statements

1. Write a program to check whether the given number is even or odd
2. Write a program to read marks of a student and display the corresponding grade
3. Write a program to find the largest element among the given numbers (multi-way if-elif-else statements.)

Module-4: Looping/Iterative Statements

1. Implement the following programs using while loop and for loop

   i. Display all prime numbers up to n. ii. Print the nth multiplication table.

   iii. Print different patterns using loops:

2. Demonstrate the following control transfer statements in Python with suitable examples.
   i. break ii. continue iii. Pass

Module-5: Functions

1. Write a function to find the multiplication of two numbers and demonstrate the usage of parameters and arguments of a function.
2. Write a program to define a function using default arguments.
3. Demonstrate lambda functions in Python with suitable example programs.

Module-6: Strings

1. Write a program to manage and analyse customer feedback using string operations include:

   i. Create feedback with name, email, comment ii. Collecting feedback.
   iii. Normalizing feedback (e.g., removing extra spaces, converting to lowercase).
   iv. Extracting key in formation (e.g., names, email addresses, and comments).
   v. Searching for keywords. vi. Replacing certain words.
   vii. Formatting feedback for display. viii. Summarizing feedback.

Module-7: Lists:

1. Write a program to create a list and perform the following operations:
   i. + ii. * iii. slicing iv. del
2. Inventory Management: You have a list of items in your warehouse along with their quantities. Write a program to find out which items are low in stock (quantity less than 10). (use only comprehensions)
3. Employee Performance: You have a list of employee names and their corresponding performance scores. Write a program to sort the list based on the performance scores in descending order.
4. Sales Analysis: you have a list of sales figures for the past week. Write a program to find the total sales, the highest sale, the lowest sale, and the average sale.
5. Write a program to calculate the length of each element in a list using map function in python.

Module-8: Tuples

1. Write a program to return the top n's most frequently occurring chars and their respective counts. e.g. string=aaaaaabbbbcccc, n=2 should return [(a 6) (b 4)].
2. Write a program to create n iterables of varied sizes and group the values using zip function in python.
3. Student Information: Write a program to create a list of tuples where each tuple contains the student ID, name, and grade and find the student with the highest grade.

4. Course Enrolment: Write a program to create list of tuples where each tuple contains the course ID and the number of students enrolled and find the total number of students enrolled across all courses
5. Faculty Information: Write a program to create a list of tuples where each tuple contains the faculty ID, name, and department and find all faculty members in a given department.
6. Library Book Tracking: You have a list of tuples where each tuple contains the book ID, title, and number of copies available. Write a program to find all books with fewer than 5 copies available.

Module-9: Sets & Dictionaries

1. Write a program to create two sets and perform the following operations:

i. Union ii. Intersection iii. Difference iv. Asymmetric Difference

2. Write a program to generate a dictionary that contains numbers (between 1 and n) in the form of (x,x*x).
3. Write a program to implement a shopping cart where you can add items with their prices and quantities, and then calculate the total cost.
4. Banking System: Write a program to create dictionary with customer name and balance and retrieve the balance for a given customer, deposit a specified amount into a customer's account, Withdraw a specified amount from a customer's account if sufficient balance is available, Transfer a specified amount from one customer's account to another's and Remove a customer from the bank's system.

Module-10: Classes and objects

1. Create a class to represent menu items in a restaurant with attributes like name, price, and category. Implement methods to display menu details and calculate the total cost of a selected list of items.
2. Write a program to read 3 subject marks and display pass or failed using class and object.

Module-11: Files

1. Write a program to copy the contents of a file to another file.
2. Write a program to compute the number of characters, words and lines in a file.

Case Study:

Select any one application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

1. Design a Python program to manage inventory, process orders, and handle customer information for an online store.
2. Create a library system to track books, manage member accounts, and handle book loans and returns.
3. Implement a personal finance management tool to track income, expenses, and generate monthly reports.
4. Create a system to manage student grades, calculate GPAs, and generate academic reports.

5. Develop a Python-based movie recommendation system using collaborative filtering.
6. Build a to-do list application that allows users to add, update, delete, and prioritize tasks..
7. Build a patient management system to handle appointments, medical records, and billing.
8. Design a digital menu for a restaurant with options for ordering, bill generation, and inventory management.
9. Implement a fitness tracking application to log workouts, track progress, and suggest exercise plans.
10. Build a secure online voting platform with user authentication and vote tallying.
11. Create a program to manage recipes, including ingredients, instructions, and nutritional information.
12. Build a travel planning application to manage itineraries, bookings, and budget tracking.
13. Develop an e-learning application with course creation, student enrolment, and progress tracking.
14. Build a car rental management system to handle bookings, returns, and vehicle maintenance.
15. Build a budget planning application to track expenses, income, and financial goals.
16. Develop a cryptocurrency tracking application to monitor prices, market trends, and portfolio performance.

Reference Books:

1. Python Programming: Using Problem Solving Approach by Reema Theraja , Oxford publications
2. Ashok N Kamthane, Amit Ashok Kamthane, Programming and Problem Solving with Python, 1st Edition, McGraw Hill Education (India), 2018.

Web References:

1. https://www.python.org/doc/
2. https://www.w3schools.com/python/python_reference.asp
3. https://thepythonguru.com/
4. https://www.programiz.com/python-programming

WEB TECHNOLOGIES LAB

I Year B. Tech. II semester

[Common to CSE, CSE (AI&ML), IT]

Course Code: 24CT11RC08

L	T	P	C
0	0	3	1.5

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

CO1: Develop professional web pages for an application using HTML elements like lists, navigations, tables, various form elements, and embedded media, including images, audio, and video. (L3)

CO2: Build inline, internal, and external CSS to enhance the visual presentation of web pages, including layout, fonts, colors, and backgrounds. (L3)

CO3: Utilize JavaScript to develop interactive HTML web pages and validate form data. (L3)

CO4: Apply JavaScript functions and events to enhance user interaction and provide a dynamic user experience. (L3)

CO5: Experiment with XML files to store and display structured data and write a DTD to perform well-formed and validated XML documents. (L3)

Module-1:

Design a web page using Basic HTML Elements – Structure of HTML, Headings, Paragraph, Division and Span, List, Link, Image, Table, Form, Input, Iframe, and Media Elements.

Module-2:

Design the following static web pages required for an online book store website.

1. **HOME PAGE:**
   i. The static home page must contain three frames.

   Top frame: Logo and the college name and links to the Home page, Login page, Registration page, Catalogue page, and Cart page (the description of these pages will be given below).

   Left frame: At least four links for navigation, which will display the catalogue of respective links.For e.g.: When you click the link "CSE" the catalogue for CSE Books should be displayed in the Right frame.

   Right frame: The pages to the links in the left frame must be loaded here. Initially, this page contains a description of the website.

   ii. Add media content in a frame using audio, video, and iframe elements to the Home page of an online bookstore website. The page should include:

   iii. Audio Element: An audio player to play background music or a podcast.

   iv. Video Element: A video player to display a promotional video about the bookstore.

v. Iframe Element: An iframe to embed a Google Maps location of the bookstore.

2. **LOGIN PAGE**
3. **CATALOGUE PAGE:** The catalogue page should contain the details of all the books available on the website in a table: The details should contain the following:

   i) Snapshot of Cover Page. ii)Author Name. iii)Publisher.

   iv) Price. v)Add to cart button.
4. **REGISTRATION PAGE:** Create a "registration form "with the following fields

   i) **Username:** Text input field ii) **Password:** Password input field

   iii) **Email:** Email input field iv) **Phone number:** Text input field

   v) **Birthday:** Date picker vi) **Favorite Color:** Color picker

   vii) **Gender:** Select an element with options (Male, Female, Other).

Module-3:

Develop and demonstrate the usage of inline, internal, and external style sheets using CSS.

Module-4:

Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (colour, bold, and font size).

Module-5:

Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problems:

1. Input: Click on the Display Date button using onclick( ) function

   Output: Display the date in the textbox.
2. Input: A number n obtained using prompt

   Output: Factorial of n number using alert.
3. Input: A number n obtained using prompt

   Output: A multiplication table of numbers from 1 to 10 of n using alert.
4. Input: A number n obtained using prompt and add another number using confirm

   Output: Sum of the entire n numbers using alert.

Module-6:

Write JavaScript to validate the following fields of the Registration page.

1. First Name (Name should contain alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters in length).
3. E-mail ID (should not contain any invalid and must follow the standard pattern name@domain.com)
4. Mobile Number (The phone number should contain 10 digits only).

5. Last Name and Address (should not be Empty).

Module-7:

Write an HTML page including any required JavaScript that takes a number from the text field in the range of 0 to 999 and shows it in words. It should not accept four and above digits, alphabets, and special characters.

Module-8:

Create an 'Employee' class extending from a base class 'Person'. The task should include:

1. Create a class 'Person' with attributes 'name' and 'age' and add a constructor to initialize these values.
2. Create a class 'Employee' that extends 'Person' and includes an additional attribute 'role,' with a constructor to initialize 'name,' 'age,' and 'role.'

Module-9:

Create an array of objects representing book details in an online bookstore. Each object should include the book title, author, genre, and price. Render the details of the books on the page using this array, demonstrating array creation, accessing array elements, and applying array methods.

Module-10:

Write an XML file which will display the Book information which includes the following:

a) Title of the Book b) Author Name c) ISBN Name
d) Publisher Name e) Edition f) Price

Write a Document Type Definition (DTD) to validate the above XML file.

Case Study:

Select any one practical application mentioned below.

Note: A report has to be submitted by every student at the end of the semester, including design, coding, output, etc.

1. Portfolio Website: Create a personal portfolio website showcasing projects, skills, and experience using HTML, CSS, and JavaScript for interactive features.
2. Responsive Blog: Design a responsive blog template with multiple pages, utilizing HTML for structure, CSS for styling, and JavaScript for dynamic content.
3. E-commerce Site: Develop a basic e-commerce website with product listings, shopping cart functionality, and user authentication using HTML, CSS, and JavaScript.
4. Quiz Game: Create an interactive quiz game with multiple-choice questions, scoring, and feedback using HTML, CSS, and JavaScript for functionality.

5. To-Do List: Develop a to-do list application allowing users to add, edit, and delete tasks, with data persistence using JavaScript local storage.
6. Interactive Resume: Design a dynamic resume with sections for education, experience, and skills, including interactive elements powered by JavaScript.
7. Photo Gallery: Create an interactive photo gallery with image lightbox effects, using CSS for styling and JavaScript for functionality.
8. Recipe Finder: Build a recipe search application that fetches and displays recipes from an API, using JavaScript for dynamic content updates.
9. Expense Tracker: Develop an expense tracker app to log and categorize expenses, with data visualization using charts and graphs created with JavaScript.
10. Event Countdown: Create a countdown timer for upcoming events with HTML structure, CSS styling, and JavaScript for the timer logic.
11. Contact Form: Create a contact form with validation and submission handling using JavaScript, styled with CSS to match the site's theme.
12. Memory Game: Develop a memory matching game with cards that flip and match, using CSS for design and JavaScript for game logic.
13. Interactive Story: Design an interactive storytelling website where users make choices that affect the narrative, using JavaScript for decision logic.
14. Fitness Tracker: Build a fitness tracker to log workouts and track progress, with dynamic charts and data visualization using JavaScript.
15. Task Manager: Create a task manager application with features for adding, editing, and prioritizing tasks, using JavaScript for interactivity.
16. Interactive Timelines: Design interactive timelines to visualize historical events or project milestones, using HTML, CSS, and JavaScript.
17. Product Landing Page: Develop a product landing page with animations and interactive elements, using CSS for styling and JavaScript for effects.
18. Calculator: Build a fully functional calculator with basic arithmetic operations, using JavaScript for logic and CSS for design.
19. Job Board: Develop a job board where users can post and search for jobs, with dynamic content updates using JavaScript.
20. Interactive Form: Create a multi-step interactive form with validation and progress indicators, using JavaScript for form handling and CSS for styling.

References:

1. Internet and Web Technologies by Raj Kamal, Tata McGraw-Hill.
2. Programming the World Wide Web by Robert W. Sebesta, Pearson Education.

3. An Introduction to Web Design and Programming by Paul S Wang, Sanda S Katila, 1st Edition, Cengage Learning.

Web Links:

1. https://onlinecourses.swayam2.ac.in/nou24_cs18/preview
2. https://infyspringboard.onwingspan.com/en/app/toc/lex_17739732834840810000_shared /overview (HTML5)
3. https://infyspringboard.onwingspan.com/en/app/toc/lex_18109698366332810000_shared /overview (JavaScript)

DISCRETE MATHEMATICAL STRUCTURES

II Year B. Tech. I Semester
[Common to CSE, CSE (AI&ML), IT]

Course Code: 24BM11RC05

L	T	P	C
3	0	0	3

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

CO1: **Interpret** relations through ordered pairs, mapping diagrams and tables [L5].

CO2: **Analyze** mathematical arguments using logical connectives and quantifiers and verify the validity of arguments using propositional logic [L4].

CO3: **Define** the main concepts related to primality, divisibility, congruences [L1].

CO4: **Apply** combinatorial principle and techniques to solve counting problems [L3].

CO5: **Classify** precise and accurate mathematical definitions of objects in graph theory [L2].

UNIT-I: **10 Lectures**

SET THEORY AND RELATIONS : Sets - Operations on Sets, Principle of Inclusion – Exclusion - Relations - Properties, Operations, Partition and Covering, Transitive Closure, Equivalence, Compatibility and Partial Ordering, Hasse Diagrams – Functions – Bijective Function – Composition of Functions – Permutation Function - Recursive Function - Lattice and its Properties.

(Sections 2.1, 2.3, 2.4, 2.6, 4-1.1 to 4-1.3 of Text book 1)

UNIT-II: **10 Lectures**

MATHEMATICAL LOGIC: Propositional Calculus: Statements and Notations, Logical Connectives, Truth Tables, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications, Normal Forms, Rules of Inference for Statement Calculus, Consistency of Premises, Indirect methods of proof. Predicate Calculus: Predicates, Predicative Logic, Statement Functions, Variables and Quantifiers, Free and Bound Variables, Inference Theory for Predicate Calculus.

(Sections 1.1 to1.6 of Chapter 1 of Text book 1)

UNIT-III: **10 Lectures**

MATHEMATICAL INDUCTION AND NUMBER THEORY: Mathematical Induction, Number Theory: Properties of Integers, Division Theorem, Greatest Common Divisor, Euclidean Algorithm, Least

Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic, Solving congruence, Fermat, Euler and the Chinese Remainder Theorems.

(Sections 5.1, 4.1, 4.2, 4.3, 4.4 of Text book 3)

UNIT-IV: **10 Lectures**

RECURRENCE RELATIONS: Basics of Combinatorics -Function of Sequences, Formulation of Recurrence Relations, Generating Functions, Partial Fractions, Calculating Coefficient of Generating Functions, Solving Recurrence Relations - Substitution, Generating Functions and Method of Characteristic roots.

(Sections 3.1 – 3.6 of Text book 2)

UNIT-V: **10 Lectures**

GRAPH THEORY : Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring – Trees - Spanning Trees – DFS and BFS algorithms - Minimum Spanning Trees – Kruskal's and Prim's algorithms. (Sections 10.1 - 10.8, 11.4 - 11.5 of Text book 3)

Text Books:

1. **J. P. Tremblay** and **P. Manohar**, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill.
2. **J. L. Mott, A. Kandel and T. P. Baker**, Discrete Mathematics for Computer Scientists and Mathematicians, 2nd Edition, Prentice Hall of India.
3. **K. H. Rosen**, Discrete Mathematics and its Applications with Combinatorics and Graph Theory, 7th Edition, Tata McGraw Hill.

Reference Books:

1. **Bernand Kolman, Robert C. Busby and Sharon Cutler Ross**, Discrete Mathematical Structures, PHI.
2. **S. K. Chakraborthy and B. K. Sarkar**, Discrete Mathematics, Oxford, 2011.
3. **Seymour Lipschutz and Marc Lars Lipson**, Theory and Problems of Discrete Mathematics, Schaum's Outline Series, 3rd Edition, McGraw Hill.
4. **C. L. Liu and D. P. Mohapatra**, Elements of Discrete Mathematics-A Computer Oriented Approach, 3rd Edition, Tata McGraw Hill.
5. **Ralph. G. Grimaldi**, Discrete and Combinatorial Mathematics, Pearson Education, New Delhi.

COMPUTER ORGANIZATION

II Year B. Tech. I semester
[Common to CSE, IT]

L	T	P	C
3	0	0	3

Course Code: 24CT11RC11

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

- **CO1:** **Demonstrate** the design of major components of a basic computer such as register, transfer Micro-operations and arithmetic logic shift unit. (L2)
- **CO2:** **Classify** various instruction codes, timing & control unit and understand the design of CPU, Micro-programmed control Unit. (L2)
- **CO3:** **Make Use of** register organization, instruction formats, and addressing modes to understand the working of central processing units (CPUs) and 8086 microprocessors. (L3)
- **CO4:** **Interpret** different ways of communication with I/O devices and parallel processors. (L2)
- **CO5:** **Outline** the performance of various memory modules and to comprehend computer arithmetic algorithms. (L2)

UNIT-I **08 Lectures**

Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro-operations, Shift Micro operations, Arithmetic Logic Shift Unit.

UNIT-II **10 Lectures**

Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator Logic.

Micro Programmed Control: Control Memory, Address Sequencing, Micro program Example, Design of Control Unit.

UNIT-III **12 Lectures**

Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes.

Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC).

8086 Microprocessor: Family Overview, Internal Architecture.

UNIT-IV **10 Lectures**

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline.

Input/Output Organization: Peripheral Devices, I/O interface, Asynchronous data transfer, Modes of transfer, priority Interrupt, Direct memory access.

UNIT-V **10 Lectures**

Computer Arithmetic: Addition and Subtraction with Signed-Magnitude Data, Booth Multiplication Algorithm.

Memory Organization: Memory Hierarchy, Main memory, Associative Memory, Cache Memory.

Textbooks:

1. Computer System Architecture, M. Morris Mano, Prentice Hall of India Pvt. Ltd., Third Edition, Sept.2008**.**
2. Microprocessors and Interfacing, D. V. Hall, TMGH, 2nd Edition 2006.

Reference Books:

1. Computer Architecture and Organization, William Stallings, PHI Pvt. Ltd., Eastern Economy Edition, Sixth Edition, 2003.
2. Computer Organization and Architecture, Linda Null, Julia Lobur, Narosa Publications ISBN81- 7319-609-5.
3. Computer System Architecture‖, John. P.Hayes.

Web References:

1. NPTEL :: Computer Science and Engineering - NOC:Computer architecture and organization

DATA STRUCTURES

II Year B. Tech. I Semester

[Common to CSE, CSE (AI&ML), IT, ECE]

L	T	P	C
3	0	0	3

Course Code: 24CT11RC09

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

CO1: **Outline** the concept of ADT, **identify** suitable data structures to solve problems, and experiment with different searching & sorting techniques using arrays. (L2,L3)

CO2: **Develop** and **analyse** algorithms for stacks, queues, and priority queues. (L3,L4)

CO3: **Model** linked list representations for various applications. (L3)

CO4: **Develop** and **analyse** algorithms for Binary Trees and Binary Search Trees. (L3,L4)

CO5: **Summarize** the concepts of Graph representation, graph traversals and hashing. (L2)

UNIT-I **10 Lectures**

Data Structures: Definition, Classification of Data Structures, Operations on Data Structures, Abstract Data Type (ADT), Introduction to array ADT.

Searching: Linear search, Binary search.

Sorting: Insertion sort, Selection sort, Exchange (Bubble sort, quick sort), distribution (radix sort)

UNIT-II **10 Lectures**

Stacks: Introduction to Stacks, Array Representation of Stacks, Operations on Stacks, Applications-Infix to Postfix Conversion, Evaluating Postfix Expressions.

Queues: Introduction to Queues, Representation of Queues-using Arrays, Implementation of Queues-using Arrays, Circular Queues.

Priority Queue: model, simple implementation, Binary Heap-structure property, heap order property, heap operations, Heap sort.

UNIT-III **12 Lectures**

Linked List: Introduction, Single linked list, Representation of Linked list in memory, Operations on Single Linked list-Insertion, Deletion, Search and Traversal, Reversing Single Linked list, Double Linked list, Circular Linked list.

Applications on Single Linked list: Stacks and queues using linked list, Polynomial Expression Representation, Sparse Matrix Representation using Linked List

UNIT-IV **10 Lectures**

Introduction: Terminology, Representation of trees

Binary Trees: The ADT, Properties of binary trees, Binary tree Representations, Binary tree Traversals: in order traversal, preorder traversal, post order traversal

Binary Search Trees: Definition, Searching a BST, Insertion into a BST, Deletion from a BST

Efficient Binary Search Tree: AVL Trees.

UNIT-V **8 Lectures**

Graphs: Basic Concepts, Representations of Graphs-Adjacency Matrix and using Linked list, Graph Traversals (BFT & DFT), Hashing-Introduction, static hashing, hashing functions, overflow handling techniques.

Text Books:

1. Fundamentals of Data Structures in C, 2nd Edition, Horowitz, Sahin, Universities Press.
2. Data Structures and algorithm analysis in C, 2nded, Mark Allen Weiss.

Reference Books:

1. Data Structures Using C. 2nd Edition, ReemaThareja, Oxford.
2. Data Structures: A PseudoCode Approach, 2/e, Richard F.Gilberg, Behrouz A. Forouzon, Cengage.

Web References (e-Resources):

1. https://onlinecourses.swayam2.ac.in/cec25_ma15/preview.

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

II Year B. Tech. I semester
[Common to CSE, CSE (AI&ML), IT]

Course Code: 24CT11RC10

L	T	P	C
3	0	0	3

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

CO1: Explain the fundamental concepts of Object-Oriented Programming and Java features. (L2)
CO2: Apply inheritance and polymorphism concepts to develop reusable and maintainable java applications. (L3)
CO3: Utilize exception handling and multithreading techniques to enhance application reliability. (L3)
CO4: Build interactive GUI applications using Java Swing and event-driven programming. (L3)
CO5: Construct data handling mechanisms using Java I/O and develop JDBC applications to perform CRUD operations with MySQL. (L3)

UNIT-I **12 Lectures**

An Overview of Java: Object-Oriented Programming- OOP Principles, The Java Buzzwords, A First Simple Program, Lexical Issues, Data Types, Variables, Arrays, Operators, Control Statements.

Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, A Closer Look at Methods and Classes - Introducing Methods, Constructors, this Keyword, Garbage Collection, Overloading Methods, Using Objects as Parameters, Understanding static, Introducing final, Introducing Nested and Inner Classes, Using Command- Line Arguments.

UNIT-II **10 Lectures**

Inheritance and Polymorphism: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, Constructor invocation- Execution of Constructors, Method Overriding, Run-time polymorphism - Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.

Packages and Interfaces –Packages and Member Access, Importing Packages, Interfaces, Use static Methods, String Handling.

UNIT-III **10 Lectures**

Exception Handling: Exception-Handling Fundamentals, Exception Types, Using try and catch, Multiple catch Clauses, Nested try Statements, Throw, Throws, Finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions.

Multithreaded Programming: The Java Thread Model, Creating a Thread and Multiple Threads, Thread States, Thread Properties, Thread Priorities, Synchronization, Inter thread Communication.

UNIT-IV **10 Lectures**

Applets and Event Handling: Applets- A Simple Applet, The Applet HTML tag and its Attributes, Two Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Handling Mouse Events, Handling Keyboard Events, Adapter and Inner Classes.

Introducing GUI Programming: Using AWT Controls, Layout Managers, and Menus, Introducing Swing.

UNIT-V **8 Lectures**

Input/Output: Exploring java.io, The Stream Classes, The Byte Streams, The Character Streams, Serialization.

Java Database Connectivity (JDBC): JDBC Introduction, JDBC Architecture, Java Database Connectivity with MySQL, CRUD operations.

Textbooks:

1. Java The Complete Reference (11th ed.), Herbert Schildt (2018), McGraw Hill. ISBN: 978-1260440232.
2. Object-Oriented Programming with Java (8th ed.), Balagurusamy, E (2019). McGraw Hill.

Reference Books:

1. Core Java Volume I – Fundamentals (8th ed.), Horstmann, Gary Cornell (2018), Pearson. ISBN: 978 813171945-9.
2. Java How to Program, Early Objects Version (11th ed.), Deitel, P., & Deitel, H. (2017). Pearson.

Web References:

1. https://onlinecourses.nptel.ac.in/noc20_cs58/preview
2. https://docs.oracle.com/javase/tutorial/
3. https://www.udemy.com/course/java-tutorial/
4. https://www.edx.org/learn/java/universidad-carlos-iii-de-madrid-introduction-to-javaprogramming-starting-to-code-in-java
5. https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc

OPERATING SYSTEMS

II Year B. Tech. I semester

[Common to CSE, IT]

Course Code: 24CT11RC13

L	T	P	C
3	0	0	3

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

CO1: Classify operating system types, concepts, and core services to understand their role in managing hardware and software resources. (L2)

CO2: Summarize the impact of scheduling policies and interprocess communication on system performance in multitasking environments. (L2)

CO3: Make use of synchronization mechanisms and deadlock handling strategies to ensure safe and efficient concurrent process execution. (L3)

CO4: Apply memory management techniques like paging, segmentation, and virtual memory to optimize resource utilization and system performance. (L3)

CO5: Utilize file system structures, allocation methods, and disk scheduling algorithms to enhance data storage and retrieval efficiency. (L3)

UNIT-I **10 Lectures**

Introduction: Operating System's role, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security.

System Structures: Operating-System Services, System Calls, Types of System Calls, Operating System Structure, Operating-System Generation, System Boot.

UNIT-II **10 Lectures**

Process Concept: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication.

Multithreaded Programming: Overview, Multithreading Models, Threading Issues.

Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiple Processor Scheduling.

UNIT-III **12 Lectures**

Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors.

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

UNIT-IV **8 Lectures**

Memory-Management Strategies: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.

Virtual-Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing.

UNIT-V **10 Lectures**

File System: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing.

Implementing File-Systems: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

Mass-Storage Structure: Overview of Mass-Storage Structure, Disk Structure, Disk Scheduling, Swap-Space Management, RAID Structure.

Text Books:

1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 9th Edition, Wiley, 2018, ISBN: 978-1118063330.
2. Operating Systems: Internals and Design Principles, William Stallings, 9th Edition, Pearson, 2018, ISBN: 978-1292214290.

Reference Books:

1. Modern Operating Systems, Andrew S. Tanenbaum, Herbert Bos, 5th Edition, Pearson, 2022.
2. Operating Systems: A Concept-Based Approach, Dhananjay M. Dhamdhere, 3rd Edition, McGraw Hill, 2017.
3. Operating Systems: A Design-Oriented Approach, Charles Crowley, 1st Edition, McGraw Hill, 2019.

Web References:

1. https://nptel.ac.in/courses/106106144.
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-828operating system- engineering-fall-2012/.
3. https://cs140.stanford.edu

DATA STRUCTURES LAB

II Year B. Tech. I semester

[Common to CSE, CSE (AI&ML), IT, ECE]

Course Code: 24CT11RC14

L	T	P	C
0	0	3	1.5

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

CO1: Apply different Searching and Sorting Techniques using arrays. (L3)

CO2: Experiment with different linear data structure concepts using stacks and Queues. (L3)

CO3: Develop linear data structure models using various Linked lists. (L3)

CO4: Build Binary Search Tree & AVL tree and examine their traversals. (L3)

CO5: Apply DFS and BFS graph traversal techniques. (L3)

Implement the following programs with either C/C++/JAVA/Python

Module-1: Searching

1. Write a program that use non recursive functions to perform linear search for a Key value in a given list.
2. Write a program that use non recursive functions to perform Binary search for a Key value in a given list.

Module-2: Sorting

1. Write a program that implement Bubble sort, to sort a given list of integers in ascending order.
2. Write a program that implement Selection sort, to sort a given list of integers in ascending order.
3. Write a program that implement Insertion sort, to sort a given list of integers in ascending order.

Module-3: Efficient Sorting

1. Write a program that implement Quick sort, to sort a given list of integers in ascending order.

Module-4: Stack & Queue

1. Write a program that implement stack (its operations) using arrays.
2. Write a program that implement Queue (its operations) using arrays.

Module-5: Singly Linked List

1. Write a program that uses functions to create and perform operations on singly linked list.

Module-6: Double Linked List

1. Write a program that uses functions to create and perform operations on double linked list.

Module-7: Circular Linked List

1. Write a program that uses functions to create and perform operations on circular linked list.

Module-8: Binary Search Trees

1. Write a program to Create a Binary Search Tree and Perform insertion and deletion operations.

Module 9: AVL tree

1. Write a program to Build an AVL tree and perform insertions.

Module-10: Graphs

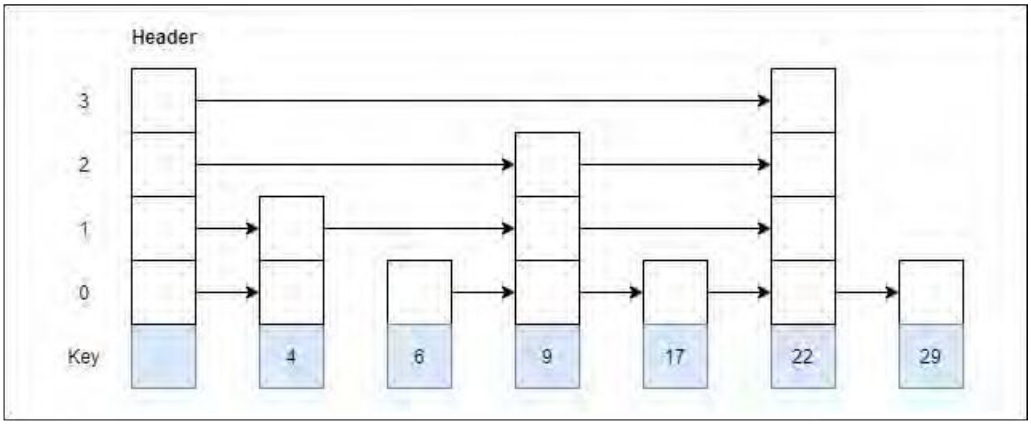
1. Write a program to implement Depth First Search
2. Write a program to implement Breadth First Search

Case Study: Select any five practical applications mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes requirements, design, coding, and output with testing results of a real example.

1. Demonstrate to convert an infix expression into a postfix expression and evaluate to find the result.
2. Demonstrate to convert an infix expression into a prefix expression
3. Demonstrate a queue using Linked List and Stack.
4. Demonstrate a sparse matrix using array and linked list
5. Create a skip list, to insert these following keys in the empty skip list.
    a. 6 with level 1.
    b. 29 with level 1.
    c. 22 with level 4.
    d. 9 with level 3.
    e. 17 with level 1.
    f. 4 with level 2.

Implement all basic operations of skip list and demonstrate with examples. Skip list structure is shown below for reference.



6. Given an array representation of min Heap, convert it to max Heap and then apply Heapsort concept to display the data in decreasing order.

Input: arr[ ] = {3, 5, 9, 6, 8, 20, 10, 12, 18, 9}

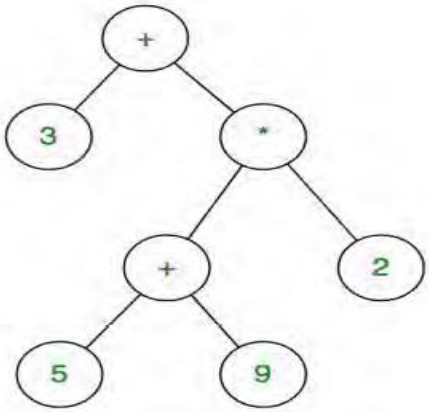
7. Make use of Radix sort algorithm to sort an array by individual digits, starting with the least significant digit.

8. Model a linked list data structure to add two polynomials.

9. Design a system to manage employee records {empID, empname, dept, salary}, and implement efficient basic operations based on employee ID.

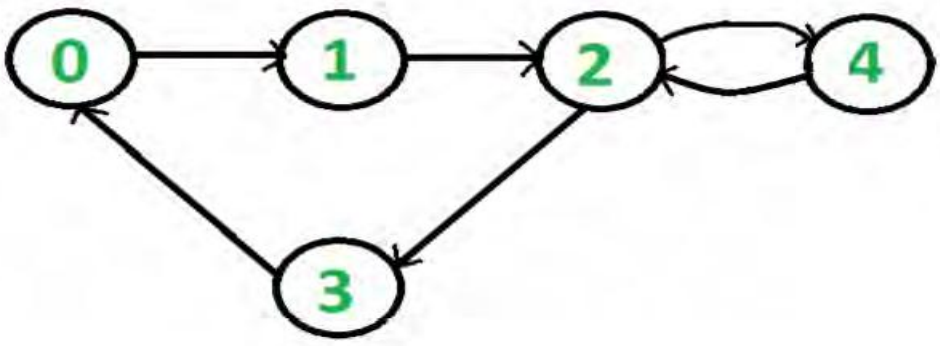
10. Construct an expression tree i.e. a binary tree in which each internal node corresponds to the operator and each leaf node corresponds to the operand.

For example: expression tree for 3 + ((5+9)*2) would be: Demonstrate with required operations to convert this above expression into corresponding prefix, and postfix expressions and evaluate the result of the expression.



11. Demonstrate topological sorting for a Directed Acyclic Graph (DAG) is a linear ordering of vertices such that for every directed edge u-v, vertex u comes before v in the increasing order a vertex with no. of incoming edges.

12. Given a directed graph, check whether the graph contains a cycle or not. Your function should return true if the given graph contains at least one cycle, else return false. For example, the following graph contains two cycles 0->1->2->3->0 and 2->4->2. Demonstrate with required operations to display the results in the form of true and the cyclic path if any. Make use of BFA concept to solve this problem.



13. Find the frequency of each character in a string using Hashing Data Structure

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

II Year B. Tech. I semester
[Common to CSE, CSE (AI&ML), IT]

Course Code: 24CT11RC15

L	T	P	C
0	0	3	1.5

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

CO1: Apply fundamental Java programming constructs to develop basic applications. (L3)

CO2: Construct object-oriented programs to develop modular and reusable software components. (L3)

CO3: Develop robust applications by implementing exception handling, multithreading, techniques to enhance reliability and efficiency. (L3)

CO4: Build interactive Graphical User Interfaces (GUIs), handle user events, and perform file operations to create user-friendly applications. (L3)

CO5: Utilize Java Database Connectivity (JDBC) to manage data storage and retrieval in Java applications. (L3)

Module-1: If Condition, Switch Case, Command Line Arguments

1. Write a Java program to determine whether a person is a Child, Teenager, or Adult based on their age. Children (age < 13) → Serve Water, Teenagers (13‑19) → Serve Badam Milk, Adults (20 and above) → Serve Coke.
2. Write a Java program that reads two command-line arguments. If both are numbers, add them. If both are strings, concatenate them. If neither, print an error message.
3. Write a Java program that converts a student mark into words. Example: Input: 56 → Output: "Fifty-Six"(Assume valid marks are 0-100 only)

Module-2: Loops, Arrays

1. Write a Java program to print the first 50 ugly numbers.
2. Write a Java program to find leap years between 1947 and 2050.
3. Write a Java program to compute the difference between the largest and smallest array values.

Module-3: Strings

1. Write a Java program to reverse each word in a given sentence while maintaining the order of words.
2. Write a Java program to check if two given strings are anagrams. Two strings are anagrams if they contain the same characters in the same frequency, but in a different

order (ignoring spaces and case sensitivity).

Module-4: Classes, Methods, Constructors

1. Write a Java program to illustrates Create a class Rectangle.The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
2. Write a Java program to create a class called Student with instance variables studentId, studentName, and grade. Implement a default constructor and a parameterized constructor that takes all three instance variables. Use constructor chaining to initialize the variables. Print the values of the variables.

Module-5: Inheritance, Polymorphism

1. Write a Java program to create a base class Vehicle with attributes like brand, model, and speed. Derive classes Car and Bike from Vehicle, each with unique attributes (e.g., Car has numDoors, and Bike has hasGear). Implement a method displayDetails() in each subclass to print specific details.
2. Write a Java program to create a base class BankAccount with methods deposit(), withdraw(), and calculateInterest(). Derive classes SavingsAccount and CurrentAccount. Override calculateInterest() in SavingsAccount to apply interest and in CurrentAccount to maintain a minimum balance condition.
3. Implement single and multi-level inheritance.
4. Demonstrate abstract classes with area calculation for shapes.

Module-6: Packages, Abstraction, Interface

1. Write a Java program to create an abstract class Medicine with a method displayLabel(). Derive classes Tablet, Syrup, and Ointment, each implementing displayLabel() with specific instructions (e.g., "Store in a cool place" for Tablet, "Shake well before use" for Syrup). Organize these classes in a package pharma.(Abstraction & Packages).
2. Write a Java program to define an interface BankAccount with methods deposit(), withdraw(), calculateInterest(), and displayBalance(). Implement this interface in SavingsAccount and CurrentAccount classes with specific logic. Place all classes in a package named banking.(Interface & Packages).
3. Write a Java program to create an abstract class Shape with an abstract method calculateArea(). Implement subclasses Circle, Rectangle, and Triangle to compute area based on user input. Use an interface Drawable with a method draw() to display the shape's type. Organize these classes in a package geometry. (Abstraction & Interface).

Module-7: Exception Handling, Multithreading

1. Write a Java program that describes exception handling mechanism with try, catch, finally, throw and throws.
2. Write a Java program to illustrate the concept of User defined Exceptions creation.
3. Write a Java program that creates threads by extending Thread class. First thread display ―Good Morning ―every 1 sec, the second thread displays ―Hello GVPW― every 2 seconds and the third display ―Welcome to JAVA‖ every 3 seconds, (Repeat the same by implementing Runnable).
4. Write a Java program to create a producer-consumer scenario using the wait() and notify() methods for thread synchronization.

Module-8: Event Handling & GUI Programming

1. Write a Java program to handle mouse and keyboard events.
2. Write a Java program to paint like paint brush in applet.
3. Write a Java program to demonstrate FlowLayout, BorderLayout, and GridLayout.

Module-9: File Handling

1. Write a Java program, which takes file/directory name from console and print whether it's a file or directory (if it exists). Also, print the details corresponding to it like is it readable, writable, where on disk is it located (absolute path), etc).
2. Write a Java program to count lines, characters, and words from a given text file as input.

Module-10: JDBC

1. Write a Java program to connect to a MySQL database using JDBC. The program should include fields such as id, name, and email in a table named users.
2. Write a Java program to implement functionality to insert a new user into the database and retrieve all records from the user's table.

Case Study:

Select any one application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

Library Management System

Objective:

1. Develop a system for managing books, borrowers, and transactions.

OOP Concepts to be Applied:

- **Encapsulation**: Book, Member, and Transaction classes with private attributes.

- **Inheritance**: Subclasses like StudentMember and FacultyMember extending Member class.
- **Polymorphism**: Overriding calculateFine() method for different types of members.
- **Exception Handling**: Handling scenarios like issuing a book when none are available.
- **JDBC**: Storing book and member details in a database.

Implementation Details:

- Book class with attributes like title, author, ISBN, and availability.
- Member class with attributes like memberID, name, contact.
- Transaction class to record book issues and returns.
- LibraryManagement class to handle book issues, returns, and database interactions.

E-Commerce Shopping Cart

Objective:

Implement a shopping cart system that allows customers to browse products, add to cart, and place orders.

OOP Concepts to be Applied:

- **Encapsulation**: Product and Order classes to store item details securely.
- **Inheritance**: ElectronicItem and ClothingItem subclasses extending Product.
- **Polymorphism**: Applying discount methods differently for different product categories.
- **Collections Framework**: Using ArrayList and HashMap for cart management.
- **JDBC**: Connecting the system with a database to store orders.

Implementation Details:

- Product class with productID, name, price, category.
- Cart class using HashMap<Product, Integer> for storing cart items.
- Order class to store order details.
- E-commerce System class to manage transactions and payment processing.

Banking System Objective:

Design a banking application to manage customer accounts, transactions, and loans.

OOP Concepts to be Applied:

- **Encapsulation**: Account details are private with public getter/setter methods.
- **Inheritance**: SavingsAccount, CurrentAccount, LoanAccount extending Account.
- **Polymorphism**: Overriding calculateInterest() for different accounts.
- **Exception Handling**: Handling insufficient balance, invalid transactions.
- **Multithreading**: Handling multiple transactions simultaneously.

Implementation Details:

- Account class with accountNo, balance, ownerName.
- Transaction class to manage deposits, withdrawals.
- Bank class with ArrayList<Account> to store customer accounts.

Online Examination System Objective:

Build an online exam portal for students to take tests and view results.

OOP Concepts to be Applied:

- **Encapsulation**: Secure student and question data.
- **Inheritance**: MultipleChoiceQuestion and DescriptiveQuestion extending Question.
- **Polymorphism**: Implementing different grading strategies.
- **Event Handling**: Capturing user inputs for answers.
- **JDBC**: Storing student results in a database.

Implementation Details:

- Question class with questionID, questionText, options[], correctAnswer.
- Exam class to manage multiple questions.
- Student class with studentID, name, marks.

Hotel Booking System Objective:

Develop a system where users can book rooms, cancel bookings, and check availability.

OOP Concepts to be Applied:

- **Encapsulation**: Private room details with public methods.
- **Inheritance**: LuxuryRoom and EconomyRoom extending Room.
- **Polymorphism**: Overriding calculatePrice() for different room types.
- **JDBC**: Managing room bookings in a database.

Implementation Details:

- Room class with roomNumber, price, availability.
- Customer class storing customerID, name, contactDetails.
- Booking class handling room reservations.

Reference Books:

1. Herbert Schildt – Java: The Complete Reference (11th Edition), McGraw Hill, ISBN: 978-1260440232.

2. Cay S. Horstmann – Core Java Volume I – Fundamentals (8th Edition), Pearson, ISBN:978-0135166307.

3. E. Balagurusamy – Object-Oriented Programming with Java (8th Edition), McGraw Hill, ISBN: 978-9353162344.

4. Paul Deitel & Harvey Deitel – Java: How to Program (11th Edition), Pearson, ISBN: 978-0134743356.

Web References (e-Resources):

1. https://docs.oracle.com/javase/tutorial/
2. https://onlinecourses.nptel.ac.in/noc20_cs58/preview

OPERATING SYSTEMS LAB

II Year B. Tech. I semester

[Common to CSE, IT]

Course Code: 24CT11RC17

L	T	P	C
0	0	3	1.5

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

CO1: Make use of Linux commands for shell programming. (L3)
CO2: Construct programs for various OS concepts and Linux commands using System Calls. (L3)
CO3: Develop and implement programs to demonstrate various CPU Scheduling Algorithms, Process Synchronization Concepts in Operating Systems and Deadlock Prevention and Avoidance Techniques. (L3)
CO4: Build programs for Memory Management Techniques and Page Replacement Algorithms. (L3)
CO5: Develop programs for various DISK Scheduling Algorithms. (L3)

Module-1: Implementation of basic Linux commands

1. Navigate to a directory, create a subdirectory, copy files into it, change permissions, and list them using various commands.
2. Create three empty text files using the touch command (e.g., file1.txt, file2.txt, file3.txt).
3. Use the cp command to copy one of the text files to another directory. If no other directory exists, create one using mkdir and then copy the file. Rename one of the text files using the mv command.
4. Create a text file in Vi that contains a list of files in your home directory. Include descriptions for each file. Use commands like :w, :q, :set number, and navigation commands to format it.
5. Modify the file using dd, yy, p, and /search commands. Save the file and quit.

Module-2: Write a C program on scheduling algorithms in Linux/Windows OS.

1. Implementation and Analysis of FCFS Scheduling Algorithm
2. Implementation and Analysis of SJF Scheduling Algorithm
3. Implementation and Analysis of Round Robin Scheduling Algorithm

Module-3: Implementation of Critical Section Problem Using Peterson's Algorithm

1. Mutual exclusion
2. Progress
3. Bounded waiting between two processes.

Module-4: Write simple shell programming in Linux OS.

1. Hello World Script

2. List Files in Directory
3. Check if a File Exists
4. Count Lines in a File
5. Sum of Numbers
6. Display Date and Time
7. Reverse a String
8. Calculate Factorial
9. Check Prime Number

Module-5: Shell programming using decision making constructs, loop constructs, file and directory manipulation

1. **Decision-Making Constructs**: Write a shell script that checks whether a number entered by the user is positive, negative, or zero. Use if, else, and elif statements for the decision making.
2. **Loop Constructs:** Write a shell script that prints the first 10 Fibonacci numbers using for loop.
3. **File and Directory Manipulation:** Write a shell script that accepts a file name from the user and checks whether it exists. If the file exists, display the number of lines in the file; otherwise, print "File does not exist."

Module-6: Simple C programs using command line arguments, system calls, library function calls, make utility in Linux OS.

1. **Command-Line Arguments: Simple Calculator**: **Description**

Write a C program that accepts command-line arguments to perform basic arithmetic operations (addition, subtraction, multiplication, division). The user should provide the operation type (+, -, *, /) and two operands.

2. **System Calls: File Copy Using open(), read(), and write()**

Write a C program that should copy a file from one location to another using system calls like open(), read(), and write().

3. **Library Function Calls: String Reversal**

Write a C program that takes a string as input and returns the reversed string using standard library functions.

Module-7: Write C programs to study deadlock avoidance and detection in Linux/Windows OS.

Module-8: C programs using system call to create processes and study parent, child process mechanism in Linux OS.

1. Simple Parent-Child Process using fork().
2. Parent Waiting for Child to Exit using wait().

Module-9: Write a C Programs to simulate free space management in Linux/Windows OS.

a) first fit b) best fit c) worst fit

Module-10: Write a C programs to create process chaining, spawning in Linux OS.

1. Process Chaining using fork()
2. Process Spawning using fork() and exec()

Module-11: Write a C programs to error handling using errno(), perror() function in Linux OS.

1. File Opening Error
2. Division by Zero
3. Memory Allocation Error
4. Invalid File Descriptor

Module-12: Write C programs to use pipe system call for interprocess communication between Parent and Child Process in Linux OS.

Module-13: Write C programs for page replacement implementing in Linux/Windows OS.

a. FIFO b. Optimal c. LRU

Module-14: Write C programs for disk scheduling algorithms in Linux/Windows OS.

a. SCAN b. SSTF c. LOOK

Case Study: Select any one case study mentioned below.

Note: At the end of the semester, each student is required to submit a report. Compare any of two operating systems such as Windows, Linux, macOS, Android, or others.

The report should include a case study covering the following topics:

1. Different operating systems implement their file systems in unique ways, depending on their architecture and design objectives. Write an overview of how file systems function across various operating systems.

2. Memory management is responsible for allocating and deallocating memory space to various programs and ensuring efficient utilization of system memory.

3. Different operating systems implement scheduling in unique ways based on their design goals and hardware requirements. Below is an overview of scheduling across various operating systems

4. Protection ensures the integrity, confidentiality, and availability of the system, preventing unintentional or intentional harm, an overview of how protection is implemented across different operating systems.

5. Each operating system handles I/O operations through a combination of system calls, device drivers, and I/O subsystems to provide seamless interaction with hardware, an overview of how different operating systems manage I/O operations input/output.

References Books :

1. Unix concepts and applications by Sumitabha Das, TMH Publications.
2. Unix programming by Stevens, Pearson Education.
3. Shell programming by Yashwanth Kanetkar.
4. Operating System Concepts by Silberschatz, and Peter Galvin.

Web References(e-Resources)

1. https://nptel.ac.in/courses/106106144
2.https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-828-operatingsystem-engineering-fall-2012/
3. https://cs140.stanford.edu

CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY USING DEVOPS

II Year B. Tech. I semester

[Skill Course: CSE (AI&ML), IT]

Course Code: 24CT11SC01

L	T	P	C
1	0	2	2

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

CO1: Illustrate the DevOps lifecycle, install and configure Git, execute Git commands, create and manage GitHub repositories, and perform file operations using version control. (L2)

CO2: Make Use of Freestyle pipeline jobs to perform upstream/downstream relationships and develop CI/CD pipelines using Declarative and Scripted approaches. (L3)

CO3: Build Java applications using Maven, configure dependencies, generate JAR files, and deploy a Spring-based web application. (L3)

CO4: Utilize Docker to execute container management commands, create custom Docker images, and containerize Python Flask to Spring Boot applications. (L3)

CO5: Demonstrate containerized applications on Kubernetes. (L2)

Module -1: Introduction to DevOps, Git installation, and basic Git commands.

1. Introduction to DevOps and DevOps Life Cycle, installation of Git and Demonstrate git Commands.

Module-2: GitHub repository creation, cloning, and file management.

1. Create a GitHub repository, clone it to your local system, create a text file (sample.txt) in the local repository with some content, stage and commit the file, push the changes to the remote repository, and verify that the file is successfully updated in the GitHub repository.

Module -3: Jenkins installation, Freestyle Projects, and Build Pipeline Plugin.

1. Jenkins Installation & Setup, Create a Freestyle Project in Jenkins to Execute a Simple Java , Html , Python Applications.
2. Create Three Freestyle Jobs, Configuring Upstream and Downstream Relationships, and Visualizing the Pipeline Using the Build Pipeline Plugin in Jenkins.

Module -4: Declarative and Scripted Pipelines in Jenkins.

1. Demonstrate Declarative Pipeline and Scripted Pipeline in Jenkins.
2. Write a Declarative Pipeline script and Configure a Pipeline Job in Jenkins.

Module -5: Jenkins CI/CD Pipelines

1. Create a Declarative Pipeline script (Jenkins file) with stages for Checkout, Build, Test, and Deploy. Configure post actions to handle success, failure, and always conditions.
2. To create Pipeline Jobs in Jenkins where one pipeline job (pipelineJob-2) triggers another pipeline job (pipelineJob-1), demonstrating the creation of upstream and downstream relationships between pipeline jobs.

Module -6: Building and Deploying Applications

1. Create a simple Java application that prints "Hello, World!" to the console. Configure the pom.xml file to include the necessary Maven plugins, build the application using Maven, and execute the resulting JAR file using Java.
2. Create a simple Java web application using the Spring Framework that displays a welcome message when accessed via a web browser. Configure the pom.xml file to include the necessary dependencies, build the application using Maven, and execute the resulting JAR file using Java.
3. Create a simple Java web application, configure the pom.xml file to include the necessary dependencies, build the application using Maven, and deploy the application using Tomcat.

Module -7: Docker installation, basic commands, and custom image creation.

1. Install Docker and practice essential Docker commands to pull images, run containers, and manage them.
2. Create a custom Docker image using a Dockerfile and run a container from it.

Module -8: Containerizing Python Flask and Spring Boot applications.

1. Develop a Python Flask web application and containerizing it using Docker.
2. Create a simple Java web application using the Spring Framework that displays a welcome message. Configure the pom.xml file to include the necessary dependencies, build the application using Maven, and deploy it into a Docker ontainer.

Module -9: Kubernetes installation, commands, and deploying containerized applications.

1. Exploring Kubernetes Commands used for interacting with a Kubernetes cluster, managing resources, and troubleshooting issues. Build and Deploy Containerized Applications on Kubernetes.

Case Study: End-to-End CI/CD Pipeline for a Web Application

Note: A report has to be submitted by every student at the end of the semester that includes the following:

Objective:

Develop a web application and implement a CI/CD pipeline to automate the build, testing, deployment, and orchestration of the application using the following tools and technologies:

1. **Version Control**: Use Git for version control and GitHub for remote repository management.
2. **Build Tool**: Use Maven to build the application.
3. **CI/CD Tool**: Use Jenkins to automate the CI/CD pipeline.
4. **Containerization**: Use Docker to containerize the application.
5. **Orchestration**: Use Kubernetes to deploy and manage the containerized application.

Reference Books:

1. Learning Continuous Integration with Jenkins: A beginner's guide to implementing Continuous Integration and Continuous Delivery using Jenkins - Nikhil Pathania ,Packt publication.
2. Jenkins 2 – Up and Running: Evolve Your Deployment Pipeline for Next Generation Automation - Brent Laster, O_Reilly publication

Web References:

1. https://www.jenkins.io/doc/book/

ENVIRONMENT SCIENCE

II Year B. Tech. I Semester

[Common to CSE, IT]

L	T	P	C
2	0	0	0

Course Code: 24BC11MC01

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

CO1: Understanding the impacts of developmental activities and mitigation measures for sustainable environment.

CO2: Categorize the importance of natural resources management for the sustenance of the life and the society.

CO3: Differentiate various forms of pollutions and their impact on the environment

CO4: Discuss elements of Sustainable Development, energy and environmental management.

CO5 : Develop technologies on the basis of ecological principles and environmental regulations which in turn helps in sustainable development

UNIT-I

Introduction: Structure and functions of Ecosystems-Ecosystems and its Dynamics-Value of Biodiversity-impact of loss of biodiversity, Conservation of bio-diversity. Environmental indicators - Global environmental issues and their impact on the ecosystems. Salient features of international conventions on Environment: Montreal Protocol, Kyoto protocol, Ramsar Convention on Wetlands, Stockholm Convention on Persistent Organic Pollutants, United Nations Framework Convention on Climate Change (UNFCCC)

UNIT-II

Natural Resources Management: Importance of natural resources management-Land as resource, Land degradation, Soil erosion and desertification, Effects of usage of fertilizer, herbicides and pesticide-watershed management. **Forest resources:** Use and over-exploitation, Mining and dams – their effects on forest ecosystems and the living beings. **Water resources:** Exploitation of surface and groundwater, Floods, droughts, Dams: benefits a nd costs. **Mineral Resources**: Impact of mining on the environment and possible environmental management options in mining and processing of the minerals. Sustainable resource management (land, water, and energy), and resilient design under the changing environment.

UNIT-III

Environmental Pollution: Local and Global Issues. Causes, effects and control measures. Engineering aspects of environmental pollution control systems. Air pollution: impacts of ambient and indoor air

pollution on human health. Water pollution: impacts water pollution on human health and loss of fresh water resources. Soil pollution and its impact on environment, Marine pollution and its impact on blue economy. Noise pollution. Air (prevention and control of pollution) Act, Water (prevention and control of pollution) Act and their amendments. Salient features of Environmental protection Act, 1986. Pollution control technologies: Wastewater Treatment methods: Primary, secondary and Tertiary.

UNIT-IV

Sustainable Development: Population and its explosion Fundamentals of Sustainable Development– Sustainability Strategies and Barriers – Industrialization and sustainable development. Circular economy concepts in waste (solid and fluid) management. **Energy and Environment:** Environmental Benefits and challenges, Availability and need of conventional energy resources, major environmental problems related to the conventional energy resources, future possibilities of energy need and availability. **Renewable Energy:** Production of Hydrogen via Water Splitting Using Photocatalytic and Photo electrocatalytic Route process of photovoltaic energy conversion, solar energy conversion technologies and devices, their principles, working and applications, disposal of solar panel after their usage. **Biomass energy:** Concept of biomass energy utilization, types of biomass energy, conversion processes, Wind Energy, energy conversion technologies, their principles, equipment and suitability in context of India.

UNIT-V

Solid waste management: Important elements in solid waste management- Waste to energy concepts. **Management of plastic waste and E-waste:** Sources, generation and characteristics of various e- and plastic wastes generated from various industrial and commercial activities; Waste management practices including onsite handling, storage, collection and transfer. E-waste and plastic waste processing alternatives. E-Waste management rules and Plastic waste management rules, 2016 and their subsequent amendments.

Text Books:

1. Environmental Studies by R. Rajagopalan, Oxford University Press
2. Bharucha, Erach (2004). Textbook for Environmental Studies for Undergraduate Courses of all Branches of Higher Education, University Grants Commission, New Delhi.
3. Base, M., Xavier, S. (2016). Fundamentals of Environmental Studies, Cambridge University Press, India 67

Reference Books:

1. Sharma, P. D., & Sharma, P. D. (2005). Ecology and environment. Rastogi Publications

2. Agarwal, K.C. 2001 Environmental Biology, Nidi Publ. Ltd. Bikaner.

3. Clark R.S. (2001). Marine Pollution, Clanderson Press Oxford (TB) 2018.

PROBABILITY AND STATISTICS

II Year B. Tech. II Semester
[Common to CSE, CSE (AI&ML), IT]

L	T	P	C
3	0	0	3

Course Code: 24BM11RC06

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

CO1: Analyze the statistical data using various graphical techniques and compute the key descriptive measures to **interpret** the distribution, spread, and shape of a dataset [L5].

CO2: Determine the mean and variance of discrete and continuous random variables and **identify** various types of discrete and continuous distributions like Binomial, Poisson and Normal [L5].

CO3: Compute the sampling distribution of means, variance, and the difference of two means and **construct** a confidence interval to estimate population mean [L3].

CO4: Draw **inference** by a test of hypothesis concerning means, variances, and proportions [L4].

CO5: Utilize the method least squares to fit a curve like straight line, parabola, exponential function and **interpret** the coefficient of correlation to find the relationship between the variables [L3].

UNIT-I: **08 LECTURES**

DESCRIPTIVE STATISTICS : Introduction to Statistics – Populations and Samples – Data Analysis – Graphical Representation – Scatter Plot – Stem and Leaf plot – Histograms - Box plot – Measures of Central tendency – Measures of Dispersion - Coefficient of variation - Standard Deviation of combination of two groups – Moments - Skewness - Kurtosis. (Sections: 1 and 6 of Chapter 1 of Text Book 1, 25.5 – 25.11 of Text Book 2)

UNIT-II: **12 LECTURES**

RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS : Review of probability and Baye's Theorem - Random Variables - Discrete and Continuous Probability Distributions - Mathematical Expectation and Variance of random variables - Mean and Variance of Linear combination of Random Variables – Binomial Distribution – Poisson Distribution – Normal Distribution.

(Sections: 1 - 3 of Chapter 3 of Text Book 1, 1 – 3 of Chapter 4 of Text Book 1, 1 – 2, 5 of Chapter 5 of Text Book 1, 2 -5 of Chapter 6 of Text Book 1)

UNIT III: **10 LECTURES**

SAMPLING DISTRIBUTIONS AND ESTIMATION: Populations and Samples – Statistic, Parameter - Sampling Distributions – Sampling Distribution of means – Central Limit Theorem – Sampling Distribution

of Difference of Two means – t- Distribution - Sampling Distribution of Variance - χ^2- distribution – F-Distribution.

Estimation: Point Estimation – Unbiased Estimator – Standard Error of Point Estimate - Interval Estimation – Estimating the mean, Proportion (Single Sample).

(Sections: 1 - 7 of Chapter 8 of Text Book 1, 1 – 7, 10 of Chapter 9 of Text Book 1)

UNIT-IV: **10 LECTURES**

TESTS OF HYPOTHESIS: Statistical Hypothesis – Null and Alternative Hypothesis – Type I and Type II errors – Level of significance – One tail and two-tail tests – Tests concerning single mean – Test concerning Two means – Paired t- test – Tests Concerning Single and Two Proportions – Tests Concerning Variances – Goodness of Fit test. (Sections: 1 – 5, 8 – 11 of Chapter 10 of Text Book 1)

UNIT-V: **08 LECTURES**

CURVE FITTING AND CORRELATION: Curve Fitting - Method of Least Squares – Straight Line - Parabola – Exponential – Power curves. Correlation – Correlation coefficient – Lines of Regression.

(Sections: 24.4 – 24.6, 25.12 – 25.14 of Text Book 2)

Text Books:

1. **Ronald E. Walpole Raymond H. Myers Sharon L. Myers, Keying Ye**, Probability and Statistics for Engineers and Scientists, 9th Edition, Pearson Publications, 2023.
2. **B. S. Grewal**, Higher Engineering Mathematics, 45th Edition, Khanna Publishers, 2024.

Reference Books:

1. **Richard A.Johnson**, "Miller & Freund's Probability and Statistics for Engineers", 8th Edition, PHI Learning India Private Limited, 2011.
2. **T. Veerarajan** - Probability, Statistics and Random Processes (3rd Edition) - Tata McGraw-Hill Education

Web References:

1. https://archive.nptel.ac.in/courses/111/105/111105090/
2. https://nptel.ac.in/courses/111102160

MANAGERIAL ECONOMICS

II Year B. Tech. II semester

[Common to CSE, ECE, IT, EEE]

Course Code: 24HM11RC01

L	T	P	C
3	0	0	3

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

CO1: Understand principles and concepts of managerial economics (L2)

CO2: Analyse concepts of demand and utility (L4)

CO3: Identify and recognize the production function and cost analysis (L3)

CO4: Classify the knowledge on market structures and pricing (L4)

CO5: Examine various business cycles (L4)

UNIT-I **08 Lectures**

Significance of Economics and Managerial Economics: Definitions of Economics- Wealth, Welfare and Scarcity definitions; Classification of Economics- Micro and Micro Economics. Definition, Nature and Scope of Managerial Economics, Differences between Economics and Managerial Economics, Main areas of Managerial Economics, Managerial Economics with other disciplines.

UNIT-II **12 Lectures**

Demand and Utility Analysis: Demand - Definition, Meaning, Nature and types of demand, Demand function, Law of demand, Assumptions and limitations. Exceptional demand curve; Elasticity of demand - Definition, Measurement of elasticity, Types of Elasticity (Price, Income, Cross and Advertisement), Practical importance of Price elasticity of demand, Role of income elasticity in business decisions, Factors governing Price Elasticity of demand. Demand forecasting methods and its uses. Utility- Meaning, Types of Economic Utilities, Cardinal and Ordinal Utility, Total Utility, Marginal Utility, The law of Diminishing Marginal Utility and its Limitations.

UNIT-III **10 Lectures**

Theory of Production and Cost analysis: Production - Meaning, Production function and its assumptions, use of production function in decision making-CVP Analysis- Calculations (Simple problems) and Limitations.

Cost analysis - Nature of cost, Classification of costs - Fixed vs. Variable costs, Marginal cost, Controllable vs. Non - Controllable costs, Opportunity cost, Incremental vs. Sunk costs, Explicit vs.

Implicit costs, Replacement costs, Historical costs, Urgent vs. Postponable costs, Escapable vs. Unavoidable costs; Economies and Diseconomies of scale.

UNIT-IV **10 Lectures**

Market Structures and Pricing : Definition of Market, Classification of markets; Salient features or conditions of different markets - Perfect Competition, Monopoly, Duopoly, Oligopoly, Importance of kinked demand curve; Monopolistic Competition. Pricing Analysis: Pricing – Significance; Different Pricing methods- Cost plus pricing, Target pricing, Marginal cost pricing, Going -rate pricing, Average cost pricing, Peak load pricing, Pricing of joint Products, Pricing over the life cycle of a Product, Skimming pricing Penetration pricing, Mark- up and Mark- down pricing of retailers.

UNIT-V **08 Lectures**

Business Cycles: Business cycles - Definition, Characteristics, Phases, Causes and Consequences; Measures to solve problems arising from Business cycles; Inflation and Deflation.

Textbooks:

1. Arya Sri, A.R., Managerial Economics and Financial Analysis, MC Graw Hill Education, New Delhi,2015.

2. Sankaran, S., Managerial Economics, Marghan Publications, 2015, Chennai.

3. J.V. Prabhakara Rao & P. Venkata Rao, Managerial Economics and Financial Analysis, Maruthi Publications

Reference Books:

1. Dwivedi, D.N., Managerial Economics, Vikhas Publishing House pvt. Ltd. 6th Edition, New Delhi,2004.

2. Dewett, K.K., Modern Economic Theory, S. Chand & Company Ltd., New Delhi, 2005.

3. Dr.B. Kuberudu & T.V. Ramana: Managerial Economics and Financial Analysis, Himalaya Publishing House

DESIGN AND ANALYSIS OF ALGORITHMS

II Year B. Tech. II Semester

[Common to CSE, CSE (AI&ML), IT]

Course Code: 24CT11RC18

L	T	P	C
3	0	0	3

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

CO1: Analyse the time and space complexities of algorithms using growth of functions. (L4)

CO2: Apply and **analyse** divide and conquer and greedy paradigm to generate efficient solutions of real-time problems. (L3,L4)

CO3: Develop and **analyse** algorithms using dynamic programming paradigm to generate optimal solutions of real-time applications. (L3,L4)

CO4: Develop and **analyse** algorithms using Backtracking paradigm to generate optimal solutions of real-time applications. (L3,L4)

CO5: Develop and **analyse** algorithms using Branch and Bound paradigm to generate optimal solutions of real-time applications. Classify polynomial and non-polynomial problems. (L3,L4)

UNIT-I **08 Lectures**

Introduction: Algorithm Criteria**,** Algorithm Specification, Performance Analysis-Space Complexity, Time Complexity, Asymptotic Notations, Amortized Analysis.

UNIT-II **12 Lectures**

Divide and Conquer: General Method, The master method for solving recurrences, Binary Search, Merge Sort, Quick Sort, Strassen's Matrix Multiplication.

The Greedy Method: The General Method, Knapsack Problem, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees, Single Source Shortest Paths.

UNIT-III **10 Lectures**

Dynamic Programming: The General Method, Multistage Graphs, All Pairs-Shortest Paths, Optimal Binary Search Trees, 0/1 Knapsack Problem, Reliability Design, The Traveling Salesperson Problem.

UNIT-IV **10 Lectures**

Backtracking: The General Method, The 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles, Knapsack Problem.

UNIT-V **10 Lectures**

Branch and Bound: The Method - Least Cost (LC) Search, The 15-puzzle, Control Abstraction for LC- Search, Bounding, FIFO Branch-and-Bound, LC Branch-and-Bound, 0/1 Knapsack Problem, Traveling Salesperson Problem.

NP-Hard and NP-Complete problems: Basic concepts, Non-Deterministic Algorithms, The Classes NP-hard and NP-Complete, Cook's Theorem.

Textbooks:

1. Introduction to Algorithms, 3rd Edition edition by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.
2. Fundamentals of Computer Algorithms‖, 2nd Edition, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Universities Press.

Reference Books:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Pearson, 4th edition
2. S. Sridhar, ―Design and Analysis of Algorithms‖, Oxford University Press.

Web References:

1. http://nptel.ac.in/courses/106101060/

DATABASE MANAGEMENT SYSTEMS

II Year B. Tech. II Semester

[Common to CSE, IT]

Course Code: 24CT11RC12

L	T	P	C
3	0	0	3

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

CO1: Explain the basic concepts of Database Management System, levels of abstraction, architecture of DBMS. (L2)

CO2: Construct the E-R model to perform conceptual design and summarize the concepts of Basic SQL, Relational Algebra, Relational Calculus to manipulate the database. (L3)

CO3: Apply Relational Model concepts to perform various operations. (L3)

CO4: Relate and Plan the concept of data planning and database design using normalization. (L2)

CO5: Make use of the ACID properties, concurrency control mechanisms, File Organizations and Indexing Techniques. (L3)

UNIT-I **10 Lectures**

Introduction: Database system, Characteristics (Database Vs File System), Advantages of Database systems, Database Applications. Describing and Storing Data in a DBMS – Relational Model, Levels of abstraction, Data Independence, Structure of a DBMS, People Who Work with Databases.

UNIT-II **10 Lectures**

Introduction to Database Design: Database Design and E-R Diagrams, Entities, Attributes and Entity Sets, Relationship and Relationship Sets, Conceptual Design With the E-R Models.

Basic SQL: Simple Database schema, data types, table definitions (create, alter), Creating tables with relationship, Implementation of key and integrity constraints different DML operations (insert, delete, update), basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions (Date and Time, Numeric, String conversion).

Relational Algebra and Calculus: Relational Algebra-Selection and Projection, Set Operation, Renaming, Joins and Division, Relational Calculus.

UNIT-III **12 Lectures**

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, The Relational Model Integrity Constraints Over Relations- Key Constraints– Foreign Key Constraints-General Constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view (updatable and non updatable), relational set operations, Complex Integrity Constraints in SQL, Triggers and Active Databases.

UNIT-IV **10 Lectures**

Schema Refinement and Normal Forms: Introduction to schema refinement, functional dependency, Reasoning abouts FDs, Normal forms-1NF, 2NF, 3NF and Boyce-codd normal form (BCNF), Properties of Decomposition- Lossless join and dependency preserving decomposition, Fourth normal form(4NF), Fifth Normal Form (5NF).

UNIT-V **10 Lectures**

Transaction Management and Concurrency Control: Transaction, ACID properties, transaction log and transaction management with SQL using commit rollback and save point, Transactions and Schedules, Concurrent Execution of Transactions, Concurrency control with locking methods-lock granularity, lock types, two phase locking for ensuring serializability, deadlocks, Concurrency control with time stamp ordering- Wait/Die and Wound/Wait Schemes Overview of Storage and Indexing - Data on External Storage, File Organizations and Indexing, Index data Structures

Textbooks:

1. Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH.
2. Database System Concepts,5/e, Silberschatz, Korth, TMH.

Reference Books:

1. Introduction to Database Systems, 8/e C J Date, PEA.
2. Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA.
3. Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Web References:

1. https://nptel.ac.in/courses/106/105/106105175/

FORMAL LANGUAGES AND AUTOMATA THEORY

II Year B. Tech. II semester

[Common to CSE, IT]

L	T	P	C
3	0	0	3

Course Code: 24CT11RC20

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

CO1: Construct a Deterministic Finite Automata (DFA) and a Non-Deterministic Finite Automata (NFA) for a given regular language and compare their structures. (L3)

CO2: Make Use of regular expressions to generate finite automata and demonstrate how to apply algebraic laws to simplify complex regular expressions. (L3)

CO3: Construct parse trees for context-free grammars and identify ambiguities in grammars and their corresponding languages. (L3)

CO4: Identify the language recognized by a Pushdown Automaton by analyzing its transitions and stack operations. (L3)

CO5: Solve computational problems by applying Turing machine constructions to model the problem-solving process. (L3)

UNIT-I **12 Lectures**

Finite Automata: Need for Automata Theory , The Central Concepts of Automata Theory, Grammar, Types of Grammars, Chomsky Hierarchy, Finite Automata -An Informal Picture of Finite Automata, Deterministic Finite Automata(DFA), Non-Deterministic Finite Automata (NFA), Design of NFA, Equivalence of DFA and NFA, Finite Automata with €- Transitions**,** Minimization of Finite Automata, Mealy and Moore Machines, Applications and Limitation of Finite Automata.

UNIT-II **8 Lectures**

Regular Expressions: Regular Expressions**,** Regular Sets, Identity Rules, Algebraic Laws for Regular Expressions**,** Finite Automata and Regular Expressions**,** Pumping Lemma**,** Closure Properties, Regular Grammar-Equivalence of Finite Automata and Regular Grammars, Converting Finite Automation to Regular Grammar.

UNIT-III **10 Lectures**

Context Free Grammars and Languages: Context Free Grammars-Definition of Context Free Grammar, Derivations using a Grammar, Leftmost and Rightmost Derivations, The Language of a Grammar, Sequential Forms, Parse Trees, Ambiguity in Grammars and Languages-Ambiguous Grammars, Removing Ambiguity from Grammars.

Simplification of Context Free Grammars-Elimination of Useless Symbols, €- productions and Unit Productions, Normal Forms for Context Free Grammars-Chomsky Normal Form and Greibach Normal Form, Pumping Lemma **,** Closure Properties, Applications of Context Free Grammars

UNIT-IV **8 Lectures**

Pushdown Automata: Definition of Pushdown Automata**-**The Formal Definition of PDA, Graphical Notation of PDA, Instantaneous Description of Push Down Automata, The Languages of PDA, Equivalence of Pushdown Automata and Context Free Grammars Conversion-From Grammar to Pushdown Automata, From PDA to Grammar, Deterministic Pushdown Automata.

UNIT-V **12 Lectures**

Turing Machine: Turing Machine as a Computational Machine, Techniques for Turing Machine Construction, Types of Turing Machines**,** Universal Turing Machine, Recursive and Recursively Enumerable Languages.

Undecidability and Computability: Decision Problems, Decidability and Decidable Languages, Halting Problem, Post Correspondence Problem**.**

Non-Deterministic Polynomial Completeness: P-Problems, NP-Problems, NP- Hard and NPComplete Problems**.**

Text Books:

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J.D.Ullman, 3rd Edition, Pearson, 2008.
2. Formal Language and Automata Theory, K.V.N.Sunitha and N.Kalyani, Pearson, 2015.
3. Theory of Computer Science-Automata, Languages and Computation, K.L.P.Mishra and N.Chandrasekharan, 3rd Edition, PHI, 2007.

Reference Books:

1. Introduction to Automata Theory, Formal Languages and Computation, Shyamalendu Kandar, Pearson, 2013.
2. Introduction to the Theory of Computation, Michael Sipser, 3rd Edition, Cengage earning, 2012.

Web References(e-Resources):

1. https://archive.nptel.ac.in/courses/106/105/106105196/
2. https://archive.nptel.ac.in/courses/106/106/106106049/

3. https://www.coursera.org/courses?query=theory+of+computation

4. https://www.classcentral.com/course/swayam-formal-language-and-automata- theory-anapplication-in-compiler-design-291835

5. https://infolab.stanford.edu/~ullman/ialc.html

ALGORITHMS LAB THROUGH C++

II Year B. Tech. II semester

[Common to CSE, CSE (AI&ML), IT]

Course Code: 24CT11RC21

L	T	P	C
0	0	3	1.5

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

CO1: Apply the object-oriented concepts such as class, object, inheritance, encapsulation and other features. (L3)

CO2: Develop the code to demonstrate merge sort, quick sort and other applications using divide and conquer technique. (L3)

CO3: Make use of Greedy and Dynamic Programming techniques to design and implement various optimal problems. (L3)

CO4: Utilize backtracking technique to solve the optimal problems like sum of subsets and Hamiltonian cycles. (L3)

CO5: Build the optimal problems like Travelling Sales Person Problem using branch and bound technique. (L3)

Module-1: Basics of C++

1. Write a C++ program to demonstrate the keywords continue and break.
2. Read a value n=10 and display 1, 2, 3, 5, 6, 8, 10 and 1, 2, 3, 4, 5.
3. Write a C++ program to create a Calculator to find addition, subtraction, multiplication, division, or remainder of two given numbers using the switch Statement as user‘s choice.
4. Write a C++ program to demonstrate call-by-value and call-by reference concepts to swap two numbers.
5. Write a C++ program to demonstrate function overloading by using function area with different combination of required parameters to calculate the area of a square, rectangle, and circle by taking suitable input values as float or integer.

Module-2: OOP concepts using C++

1. Write a C++ program using a class called Student with the following attributes within it. **student_roll, student_name, programme_name, phone_number, grade** Create n=7 Student objects and print the **student_roll, student_name, programme_name, phone_number, and grade>8.0** of these objects with suitable headings. Demonstrate the program using constructor and destructor.

2. Write a C++ program to read two integers a and b. Compute a/b and print, when b is not zero. Raise an exception when b is equal to zero.

Module-3: Inheritance using C++

1. Write a C++ program to design a superclass called Staff with details as StaffId, Name, Phone, and Salary. Extend this class by writing three subclasses namely Teaching (branch, highest_degree, no_of_publications), Technical (skills), and Contract (period). Read and display at least 3 staff objects of all three categories. Demonstrate above program by deriving subclasses using all three access modes (public, private, and protected).

Module-4: Divide-and-Conquer

1. Write a C++ program to demonstrate following sorting techniques
    a. Merge Sort
    b. Quick Sort

Read at least 15 numbers in a range of 1 to 100 and display the sorted output.

2. Write a C++ program to find the Maximum and Minimum element in the given Array

Module-5: The Greedy Method

1. Write a C++ program to find solution for job sequencing with deadlines problem.
2. Write a C++ program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.
3. Write a C++ program to perform Single source shortest path problem for a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

Module-6: Dynamic Programming

1. Write a C++ program to implement following algorithms using Dynamic Programming Concept.
    a. All-Pairs Shortest Paths problem using Floyd's algorithm
    b. 0/1 Knapsack problem

Module-7: Backtracking:

1. Write a C++ program to implement following algorithms using Backtracking Concept.
    a. Sum of subsets problem for a given set of distinct numbers
    b. Hamiltonian cycle problem.

Module-8: Branch and Bound

1. Write a C++ program to implement following algorithm using Branch and Bound Concept.
    a. Travelling Sales Person problem.

Case Study: Select any five practical applications mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

1. To Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n>= 5000 and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using CPP how the divide-andconquer method works along with its time complexity analysis: worst case, average case and best case.
2. To sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n>= 5000, and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using CPP how the divide-andconquer method works along with its time complexity analysis: worst case, average case and best case.
3. Write a C++ program to implement Strassen‘s Matrix Multiplication Algorithm using the divide and conquer technique.
4. Write a C++ program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal‘s algorithm.
5. Write a C++ program to implement reliability optimization in system design using Dynamic Programming (DP). The goal is to select a combination of components that provides the maximum system reliability while staying within a given budget.
6. Write a C++ program to implement the Optimal Binary Search Tree (OBST) algorithm using Dynamic Programming (DP). The goal is to arrange search terms based on their frequencies so that frequent queries are retrieved faster, optimizing the auto-complete feature in a search engine.
7. Write a C++ program to implement the Graph Coloring Problem using the Backtracking Algorithm for scheduling university exams. The goal is to assign the minimum number of time slots (colors) to subjects such that no two exams with common students (conflicting subjects) are scheduled at the same time.
8. Write a C++ program to implement the N-Queens Problem using the Backtracking Algorithm to arrange students in a classroom, exam hall, or conference room while maintaining social distancing. The

goal is to ensure that: No two students sit in the same row, No two students sit in the same column, No two students sit diagonally adjacent.

9. Write a C++ program to solve the 0/1 Knapsack Problem using the Branch and Bound approach. Consider a scenario where a relief truck has a fixed weight capacity, and you must select essential supplies (food, medicine, blankets, etc.) to maximize the total priority value while staying within the weight limit.

10. Write a C++ program to solve the 15-Puzzle Problem using the Branch and Bound approach. The goal is to arrange the tiles in ascending order (from 1 to 15) with the blank space (0) at the bottom-right corner by sliding tiles into the empty space using the fewest moves.

Reference Books:

1. T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press.
2. J. Kleinberg and E. Tardos, Algorithm Design, Addison-Wesley.
3. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins.
4. A. Gibbons, Algorithmic Graph Theory, Cambridge University Press.
5. Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, John Wiley.
6. R. Sedgewick, Algorithms in C (Parts 1-5), Addison Wesley.
7. M. H. Alsuwaiyel, Algorithm Design Techniques and Analysis, World Scientific.
8. Gilles Brassard and Paul Bratley, Algorithmics: theory and practice, Prentice-Hall.
9. Udi Manber, Introduction to Algorithms: A Creative Approach, Addison-Wesley.
10. Sara Baase and Allen Van Gelder, Computer Algorithms: Introduction to Design and Analysis, Addison-Wesley.

Web References:

1. https://archive.nptel.ac.in/courses/106/105/106105164
2.https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-toalgorithms-fall-2011/

DATABASE MANAGEMENT SYSTEMS LAB

II Year B. Tech. II semester

[Common to CSE, IT]

Course Code: 24CT11RC16

L	T	P	C
0	0	3	1.5

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

CO1: Build the database using SQL DDL, DML commands and make use of built-in functions to write queries. (L3)

CO2: Identify and implement different operators and Apply nested queries to solve real time problems. (L3)

CO3: Plan Queries on Joins, Aggregate functions, views. (L3)

CO4: Construct simple PL/SQL programs using control statements and exception handling methods. (L3)

CO5: Develop applications using PL/SQL including procedures, functions, cursors, packages and triggers. (L3)

Implement the following modules using ORACLE

Module-1:

Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

Module-2:

Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)

Module-3:

Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSECT Constraints

Module-4:

Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, ORDER BY, HAVING clauses

Module-5:

Queries on Views and Joins

Module-6: PL/SQL

1. Create a simple PL/SQL program which includes declaration section, executable section and exception Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
2. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

Module-7:

Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

Module-8:

Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT–IN Exceptions, and USER DEFINED Exceptions, RAISE_APPLICATION _ERROR

Module-9:

Programs development using creation of procedures, passing parameters IN and OUT of Procedures

Module-10:

Program development using creation of functions, invoke functions in SQL Statements and write complex functions

Module-11:

1. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables**.**
2. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

Case Study:

Select any one application mentioned below and Apply the Database Design steps

Note: A report has to be submitted by every student at the end of the semester

1. Accounting Package for Shops
2. Database Manager for Magazine Agency or Newspaper Agency
3. Ticket Booking for Performances

4. Preparing Greeting Cards & Birthday Cards
5. Personal Accounts - Insurance, Loans, Mortgage Payments, Etc.,
6. Doctor's Diary & Billing System
7. Personal Bank Account
8. Class Marks Management
9. Hostel Accounting
10. Video Tape Library
11. History of Cricket Scores
12. Cable TV Transmission Program Manager
13. Personal Library
14. Sailors Database
15. Suppliers and Parts Database

Reference Books:

1. Oracle: The Complete Reference by Oracle Press.
2. Nilesh Shah, "Database Systems Using Oracle‖, PHI, 2007.
3. Rick F Vander Lans, ―Introduction to SQL‖, Fourth Edition, Pearson Education, 2007.

Web References:

1. https://docs.oracle.com/en/
2. https://www.w3schools.com/sql/sql_exercises.asp
3. https://www.w3resource.com/sql-exercises/

R PROGRAMMING

II Year B. Tech. II semester

[Skill Course: IT]

Course Code: 24IT11SC01

L	T	P	C
1	0	2	2

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

CO1: Interpret basic understanding of R Programming Language and its Sessions

CO2 : Apply the basic Statistical concepts and statistical Distributions.

CO3: Summarize the concepts of Data Structures and Data Frames in R

CO4: Demonstrate the execution of Control Structures and functions in R

CO5: Utilize the Graphics in R in the form of charts and graphs.

Module-1: Introduction to R Programming and Basic Operations

1. Write an R program to take input from the user (name and age) and display the values. Also, print the version of R installation.
2. Write an R program to get the details of the objects in memory.

Module-2: Sequences, Arithmetic Operations, and Data Visualization

1. Write an R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and the sum of numbers from 51 to 91.
2. Write an R program to create a simple bar plot of five subjects' marks.

Module-3: Working with Vectors and Matrices Creation

1. Write an R program to get the unique elements of a given string and unique numbers of a vector.
2. Write an R program to create three vectors (a, b, c) with three integers each and combine them into a 3×3 matrix where each column represents a vector. Print the content of the matrix.
3. Write an R program to create a 5×4 matrix, a 3×3 matrix with labels and fill the matrix by rows, and a 2×2 matrix with labels and fill the matrix by columns.
4. Write an R program to concatenate two given matrices of the same column but different rows.
5. Write an R program to find the row and column index of the maximum and minimum values in a given matrix.

Module-4: Array Creation, Manipulation & Operations

1. Write an R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then the first row of the third array.
2. Write an R program to create a two-dimensional 5×3 array of a sequence of even integers greater than 50.
3. Write an R program to create an array using four given columns, three given rows, and two given tables and display the content of the array.

Module-5: Data Frames and File Handling

1. Write an R program to create an empty data frame.
2. Write an R program to create a data frame from four given vectors.
3. Write an R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame.
4. Write an R program to save the information of a data frame in a file and display the information of the file.

Module-6: Lists, Nested Lists and Accessing Elements

1. Write an R program to create a list containing a vector, a matrix, and another list, and give names to the elements in the list.
2. Write an R program to access the first and second elements of a given list.
3. Write an R program to remove the second element from a given list.
4. Write an R program to select the second element of a given nested list.
5. Write an R program to merge two given lists into one list.

Module-7: Factors, Ordered Data and Naming Elements

1. Write an R program to create a list named ‘s’ containing a sequence of 15 capital letters, starting from ‘E’.
2. Write an R program to assign new names "a", "b", and "c" to the elements of a given list.
3. Write an R program to find the levels of a factor of a given vector.
4. Write an R program to create an ordered factor from data consisting of the names of months.
5. Write an R program to concatenate two given factors into a single factor.

Module-8: Vector Creation, Computation & Matrix Operations

1. Write an R program to append a value to a given empty vector.
2. Write an R program to multiply two vectors of integer type and length 3.

3. Write an R program to find the Sum, Mean, and Product of a vector, ignoring elements like NA or NaN.
4. Write an R program to create a matrix from a list of given vectors.

Text books:

1) The Art of R Programming, Norman Matloff, Cengage Learning.
2) R for Everyone, Lander, Pearson.

Reference books:

1) R Cookbook, PaulTeetor, Oreilly.
2) R in Action,Rob Kabacoff, Manning

PROFESSIONAL ETHICS AND HUMAN VALUES

II Year B. Tech. II Semester
[Common to CSE, IT]

L	T	P	C
2	0	0	0

Course Code: 24HM11MC01

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

CO1: Make use of the human values, morals and ethical principles that ought to govern the engineering profession [L3]

CO2: Relate the importance of human values and professional/engineering ethics at work. [L1]

CO3: Cultivate the moral values and dispositions that engineers ought to instill and tackle ethical challenges and moral dilemmas in engineering by utilizing ethical theories and moral development concepts. [L3]

CO4: Outline the knowledge of risk assessment techniques, rights and maintain good Collegiality and loyalty to prevent occupational crime [L2].

CO5: Identify global issues and use ethical principles to navigate professional challenges [L3].

Unit - I **06 Lectures**

HUMAN VALUES: Values - Respect - Caring - Sharing - Honesty- Courage - Self-confidence-Self exploration- Self-awareness –Intentional competency - Communal Harmony-character- Spirituality.

Unit-II **06 Lectures**

PROFESSIONAL VALUES: Integrity - Discipline - Valuing time - Cooperation - Commitment - Empathy- Code of conduct - Challenges in the workplace.

Unit - III **06 Lectures**

PROFESSIONAL ETHICS: Overview - Engineering ethics –Theories of moral developments- Kohlberg Theory- Gilligan theory- Heinz dilemma.

Moral issues - Profession - Models of professional roles – Responsibility.

Unit - IV **06 Lectures**

RESPONSIBILITIES AND RIGHTS: Safety and risk - Collegiality and loyalty - Confidentiality - Occupational crime – Human rights - Employee rights - Intellectual property rights.

Unit - V **06 Lectures**

GLOBAL ISSUES: Globalization - Environmental ethics - Computer ethics - Code of ethics - Multinational corporations - Engineers as advisors in Planning and Policy making.

Text Books:

1. R.S. Nagarajan. A Textbook on Professional Ethics and Human Values. New Age International Publishers. 2006.
2. R. Subramanian. Professional Ethics. OUPIndia.2013.

Reference Books:

1. Premvir Kapoor. Professional Ethics and Human Values. Khanna Publishing House.2019.
2. B.S. Raghavan. Human Value and Professional Ethics. S. Chand Publications. 2012.
3. R.R. Gaur & Others. A Foundation Course in Human Values and Prof. Ethics. Excel Books. 2009.
4. A.N. Tripathi. Human Values. New Age International (P) Limited.2009

COMPUTER NETWORKS

III Year B. Tech. I Semester
[Common to CSE & IT]

Course Code: 24CT11RC22

L	T	P	C
3	0	0	3

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

CO1: Summarize the concepts of Computer networks and Network reference models (OSI and TCP/IP reference models) and Transmission media. (L2)

CO2: Utilize Data link layer framing techniques, Error control techniques using CRC, Data link protocols, HDLC and PPP. (L3)

CO3: Outline the Media Access control problem in a network using multiple access protocols– ALOHA, CSMA, multiplexing, Ethernet and Wireless LAN protocols. (L2)

CO4: Make use of the Network Layer Design issues, routing algorithms and Internet protocols to perform better network communication. [L2]

CO5: Apply the Transport layer protocols- TCP, UDP protocol working mechanism in Client – Server Data communication and Application layer protocols. [L3]

UNIT-I: **10 Lectures**

Introduction to Computer Networks: Network types LAN, MAN, WAN, Network Topologies, Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models, Internet History.

Physical Layer –Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and unguided media: Wireless-Radio waves, microwaves, infrared.

UNIT-II: **12 Lectures**

Data link layer: Design issues, Framing: fixed size framing, variable size framing, flow control, error control, error detection and correction codes, CRC, Checksum: idea, one_s complement internet checksum, services provided to Network Layer.

Elementary Data Link Layer protocols: simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel.

Sliding window protocol: One bit, Go back N, Selective repeat-Stop and wait protocol, Data link layer in HDLC: configuration and transfer modes, frames, control field, point to point protocol (PPP): framing transition phase, multiplexing, multi-link PPP.

UNIT-III: **10 Lectures**

Media Access Control: Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance.

Controlled Access: Reservation, Polling, Token Passing.

Channelization: frequency division multiple Access(FDMA), time division multiple access(TDMA), code division multiple access(CDMA).

Ethernet(IEEE802.3):Standard Ethernet, Fast Ethernet, Gigabit Ethernet.

Wireless LAN(IEEE802.11): Architecture and Protocol stack.

UNIT-IV: **12 Lectures**

The Network Layer Design Issues – Store and Forward Packet Switching-Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks.

Internet Protocols: IPv4, IPv6, and DHCP, Routing Algorithms-The Optimality principle- Shortest path, Flooding, Hierarchical, Distance vector, Link state, Congestion Control algorithms-Leaky bucket and Token bucket.

UNIT-V: **10 Lectures**

The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications-Transmission control protocol: TCP services TCP features- Segment- A TCP connection- windows in TCP- flow control-Error control, Congestion control in TCP.

Application Layer – World Wide Web: HTTP, Electronic Mail-Architecture, Domain Name System, SNMP.

Text Books:

1. Computer Networks — Andrew S Tanenbaum, Fifth Edition. Pearson Education/PHI
2. Data Communications and Networks – Behrouz A. Forouzan, Fifth Edition TMH.

Reference Books:

1. Data Communications and Networks- Achut S Godbole, Atul Kahate.
2. Computer Networks, Mayank Dave, CENGAGE.

Web References (e-Resources):

1. https://inl.info.ucl.ac.be/CNP3
2. https://wps.pearsoned.com/ecs_kurose_compnetw_8/
3. https://www.netacad.com/

DATA WAREHOUSING & DATA MINING

III Year B. Tech. I semester
[Common to CSE & IT]

L	T	P	C
3	0	0	3

Course Code: 24CT11RC19

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

CO1: **Illustrate** the importance of Data Mining and its functionalities. (L2)

CO2: **Summarize** various Data Preprocessing Techniques and outline concepts of data warehousing. (L2)

CO3: **Apply** various classification algorithms on real world data to perform classification, model evaluation. (L3)

CO4: **Make use of** association rule mining techniques viz. Apriori and FP Growth algorithms and analyze on frequent item sets generation. (L3)

CO5: **Outline** the various clustering methods of unsupervised learning evaluate the result. (L2)

UNIT-I **07 Lectures**

Introduction: Importance of Data Mining and Data Warehousing, Kinds of Patterns can be mined, Which Technologies Are Used, Applications, Major Issues in Data Mining

Getting to Know Your Data: Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Measuring Data Similarity and Dissimilarity

UNIT-II **12 Lectures**

Data Pre-processing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization

Data Warehousing and OLAP Technology: Basic Concepts of Data warehouse, Data Warehouse Modelling: Data Cubes and OLAP, Data Generalization by Attribute Oriented Induction.

UNIT-III **10 Lectures**

Classification: Basic Concepts, General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Bayes Classification Methods, Model Evaluation and Selection

UNIT-IV **10 Lectures**

Association Analysis: Problem Definition, Frequent Item set Generation, Rule Generation: Confident Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP- Growth Algorithm

UNIT-V **10 Lectures**

Cluster Analysis: Overview, Basics and Importance of Cluster Analysis, Clustering techniques, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Agglomerative Clustering Algorithm, DB SCAN Algorithm, Evaluation of Clustering

Textbooks:

1. Data Mining concepts and Techniques, 3/e, Jiawei Han, Michel Kamber, Elsevier, 2011.
2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

Reference Books:

1. Data Mining Techniques and Applications: An Introduction, Hongbo Du, Cengage Learning.
2. Data Mining: VikramPudi and P. Radha Krishna, Oxford Publisher.
3. Data Mining and Analysis - Fundamental Concepts and Algorithms; Mohammed J. Zaki, Wagner Meira, Jr, Oxford

Web References:

1. NPTEL course by Prof.Pabitra Mitra, http://onlinecourses.nptel.ac.in/noc17_mg24/previe
2. NPTEL course by Dr. Nandan Sudarshanam& Dr. Balaraman Ravindran) http://www.saedsayad.com/data_mining_map.html

ARTIFICIAL INTELLIGENCE

III Year B. Tech. I Semester

[IT]

Course Code: 24IA11RC01

L	T	P	C
3	0	0	3

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

CO1: Outline the basic knowledge of Artificial Intelligence and Relate the role of AI in various problem-solving methods using informed and uninformed strategies. (L2)

CO2: Utilize logic concepts, including propositional and predicate logic to formalize reasoning and problem-solving tasks. (L3)

CO3: Make use of rules in knowledge representation techniques to encode knowledge in the evolution of AI.

CO4: Extend the concepts and methods in advanced AI by including symbolic reasoning under uncertainty and Statistical Reasoning. (L2)

CO5: Construct plans for solving hard problems in playing games, could learn from its mistakes and improve its performance to shape expert systems. (L3)

UNIT-I: **10 Lectures**

Introduction to Artificial Intelligence: Introduction, Brief History, Intelligent Systems, Foundations of AI, Applications, Tic-Tac-Tie Game Playing.

Problem Solving: General Problem Solving, Characteristics of Problem, Exhaustive Searches, Heuristic Search Techniques (Hill Climbing, Best-First Search, A* Algorithm), Iterative Deepening A*, Constraint Satisfaction, Means-Ends Analysis.

UNIT-II: **10 Lectures**

Logic Concepts: Introduction, Propositional Calculus, Propositional Logic, Natural Deduction System, Axiomatic System, Semantic Tableau System in Propositional Logic, Resolution Refutation in Propositional Logic.

Predicate Logic: Representing Simple Facts in Logic, Representing Instance and is-a Relationships, Computable Functions and Predicates, Resolution.

UNIT-III: **10 Lectures**

Knowledge Representation: Approaches to Knowledge Representation, Knowledge Representation Using Semantic Network, Extended Semantic Networks for KR, Knowledge Representation Using Frames.

Advanced Knowledge Representation Techniques: Conceptual Dependency Theory, Script Structure, CYC Theory, Case Grammars, Semantic Web.

UNIT-IV: **10 Lectures**

Symbolic Reasoning Under Uncertainty: Introduction to Non-Monotonic Reasoning, Logics for Non-Monotonic Reasoning, Implementation using Depth-First-Search With Dependency-Directed Backtracking, Implementation using Breadth-First-Search.

Statistical Reasoning: Probability and Baye's Theorem, Certainty Factors and Rule-Based Systems, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic.

UNIT-V: **10 Lectures**

Game Playing: The Minimax Search Procedure, Adding Alpha-Beta Cutoffs.

Planning: An Example Domain-The blocks World, Components of a Planning System, Goal Stack Planning.

Expert Systems and Applications: Phases in Building Expert Systems, Expert System Architecture, Expert System Vs Traditional Systems, Rule-Based Expert Systems, Application of Expert Systems, List of Shells and Tools.

Textbook:

1. Elaine Rich and Kevin Knight, Tata McGraw Artificial Intelligence, -Hill Publications.
2. Saroj Kaushik, CENGAGE Learning. ISBN-13: 978-81-315-1099, ISBN 10: 81-315-10999.Artificial Intelligence.

Reference Books:

1. Artificial intelligence, structures and Strategies for Complex problem solving, - George F Lugar, 5th ed, PEA, ISBN: 978-81-317-2327-2
2. Artificial intelligence, A Modern Approach, 3rd ed, Stuart Russel, Peter Norvig,PEA,ISBN-978-93-325-4351-5

Web References:

1. https://nptel.ac.in/courses/106102220

COMPUTER NETWORKS LAB

III Year B. Tech. I semester

[Common to CSE & IT]

Course Code: 24CT11RC24

L	T	P	C
0	0	3	1.5

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

CO1: Understand the Network Establishment and communication with the uses of NS2 Simulator, Linux and Network Commands. [L2]

CO2: Experiment Data link layer framing methods, Error detection and correction codes like Checksum, CRC, and Hamming code. [L3]

CO3: Apply Network layer routing methods for Distance Vector Routing Algorithm and shortest path routing algorithm. [L3]

CO4: Make use of socket programming and Transport layer protocols TCP, UDP for data communication in a Computer Network. [L3]

CO5: Make use of Application layer protocols HTTP and FTP for data communication in a Computer Network. [L3]

Module-1:

1. Write different types of Network cables and practically implement the cross-wired cable and straight through cable using clamping tool.
2. Connect the computers in Local Area Networks.
3. Write and Execute Basic Network commands and Network configuration commands like ping, ipconfig, traceroute, nslookup, netstat, whois, arp, rarp, etc
4. Write and Execute Basic Linux commands.

Module-2:

1. Configure a Network topology using packet tracer software.
2. Simulate to find the Number of Packets Dropped using NS2 Simulator.
3. Configure subnet on Linux using ip commands. Verify connectivity with ping and trace route.

Module-3:

1. Write a program to implement the data link layer framing methods such as character stuffing and bit stuffing.
2. Write a program to implement on a data set of characters the three CRC polynomials CRC-12, CRC-16 and CRC-CCIP.

Module-4:

1. Write a program to implement Error Detection Technique using Checksum.
2. Write a program to implement Error Correction Technique using Hamming code.

Module-5:

1. Write a program to implement Stop and Wait Protocol.
2. Write a program to implement Go back-N protocol.
3. Write a program to implement Sliding window protocol.

Module-6:

1. Write a program to implement High-Level Data Link Control protocol.
2. Write a program to implement Point-to-Point protocol.

Module-7:

1. Write a program to implement Dijkstra_s algorithm to compute the Shortest path in a graph.
2. Write a program to implement Flooding routing algorithm.

Module-8:

1. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table at each node using distance vector routing algorithm.
2. Write a program to implement Link state routing algorithm.
3. Write a program for congestion control using leaky bucket algorithm

Module-9:

1. Understand and Usage of elementary socket system calls (socket (), bind(), listen(), accept(),connect(),send(),recv(),sendto(),recvfrom())
2. Write a program to implement Connection oriented service protocol (TCP).
3. Write a program to implement Connection less service protocol (UDP).

Module-10:

1. Write a program to implement HTTP protocol.
2. Write a program to implement FTP protocol.
3. Write a program to implement SMTP protocol.

Case Study: Select any one practical application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

i. Configuring routers and switches, experimenting with routing algorithms and analysing network performance under different topologies.
ii. Students analyse network protocols (e.g., TCP, UDP, HTTP, FTP) using tools like Wireshark to understand packet structures, communication flows, and protocol behaviour.
iii. Setting up IoT-based networks with sensors and actuators, evaluating wireless communication protocols (e.g., Zigbee, LoRa, MQTT), and assessing security vulnerabilities.

Reference Books:

1. Computer Networks — Andrew S Tanenbaum, Fifth Edition. Pearson Education/PHI
2. Data Communications and Networks – Behrouz A. Forouzan, Fifth Edition TMH.
3. Data Communications and Networks- Achut S Godbole, Atul Kahate
4. Computer Networks, Mayank Dave, CENGAGE

Web References (e-Resources):

1. https://inl.info.ucl.ac.be/CNP3
2. https://wps.pearsoned.com/ecs_kurose_compnetw_8/
3. https://www.netacad.com/

DATA MINING LAB

III Year B. Tech. I semester

[Common to CSE & IT]

Course Code: 24CI11RC01

L	T	P	C
0	0	3	1.5

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

CO1: Make Use of WEKA Machine Learning Tool to perform various Data Mining Functionalities. [L3]

CO2: Perform data pre-processing tasks and association rule mining on various data sets Using Weka tool and Apply data structures available in R programming to Develop R programs to perform several data analytics operations like plotting, boxplots, normalization, discretization, transformation, attribute selection, etc., on datasets. [L3]

CO3: Classify the realistic data using various classification algorithms in Weka tool and Experiment with Knowledge Flow in weka to visualize the knowledge flow of Various Data Mining Algorithms. [L2]

CO4: Develop R programs to build regression and classification models for numerical and categorical datasets and evaluate the models with appropriate performance metrics. [L3]

CO5: Identify clusters for the realistic data using K-Means Clustering algorithm in Weka tool. [L3]

Module-1: Explore machine learning tool WEKA-II

1. Downloading and/or installation of WEKA data mining toolkit.
2. Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
3. Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, classify panel, Cluster panel, Associate panel and Visualize panel)
4. Study the arff file format Explore the available data sets in WEKA. Load a data set (ex. Weatherdataset, Iris dataset, etc.)
5. Load each dataset and observe the following:
   a. List the attribute names and they types
   b. Number of records in each dataset
   c. Identify the class attribute (if any)
   d. Plot Histogram
   e. Determine the number of records for each class.
   f. Visualize the data in various dimensions

Module-2: Perform data preprocessing tasks and demonstrate performing association rule mining on data sets

1. Explore various options available in Weka for preprocessing data and apply Unsupervised filters like Discretization, Resample filter, etc. on each dataset
2. Load weather. nominal, Iris, Glass datasets into Weka and run Apriori Algorithm with different support and confidence values.
3. Study the rules generated. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated
4. Derive interesting insights and observe the effect of discretization in the rule generation process

Module-3: Demonstrate performing classification on data sets

1. Load each dataset into Weka and run 1d3, J48 classification algorithm. Study the classifier output.Compute entropy values, Kappa statistic.
2. Extract if-then rules from the decision tree generated by the classifier, Observe the confusionmatrix.
3. Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbourclassification. Interpret the results obtained.
4. Plot RoC Curves
5. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, anddeduce which classifier is performing best and poor for each dataset and justify.

Module-4: Demonstrate performing clustering of data sets

1. Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters).
2. Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
3. Explore other clustering techniques available in Weka.
4. Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain.

Module-5: Demonstrate knowledge flow application in Weka on data sets

1. Develop a knowledge flow layout for finding strong association rules by using Apriori FP Growth algorithms
2. Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm
3. Demonstrate plotting multiple ROC curves in the same plot window by using J48 and Random Forest tree

Module-6: Load the _iris. CSV‘ file and display the names and type of each column. Find statistics such as min, max, range, mean, median, variance, standard deviation for each column of data in R.

Module-7:

1. Write R program to normalize the variables into 0 to 1 scale using min-max Normalisation.
2. Generate histograms for any one variable (sepal length/ sepal width/ petal length/ petal width) and generate scatter plots for every pair of variables showing each species in differ color.
3. Generate box plots for each of the numerical attributes. Identify the attribute with the variance.

Module-8: Study of homogeneous and heterogeneous data structures such as vector, matrix, array, list, data frame in R.

Module-9: Write R Program using _apply‘ group of functions to create and apply normalization Function on each of the numeric variables/columns of iris dataset to transform them into a value around 0 with z-score normalization

Module-10:

1. Use R to apply linear regression to predict evaporation coefficient in terms of air velocity using the data given below:
   a. Air Velocity (cm/sec) 20,60,100,140,180,220,260,300,340,380
   b. Evaporation Coefficient (sqmm/sec). 0.18, 0.37, 0.35, 0.78, 0.56, 0.75, 1.18, 1.36, 1.17,1.65
2. Analyse the significance of residual standard-error value, R-squared value, F- statistic. Find the correlation coefficient 7for this data and analyse the significance of the correlation value.
3. Perform a log transformation on the _Air Velocity 'column, perform linear regression again, and analyse all the relevant values, and analyse all the relevant values

Module-11: Write R program for reading _state.x77‘ dataset into a dataframe and apply multiple regression to predict the value of the variable _murder‘ based on the other independent variables based on their correlations

Module-12: Write R program to split _Titanic‘ dataset into training and test partitions and build a decision tree for predicting whether survived or not given the description of a person travelled. Evaluate the performance metrics from the confusion matrix**.**

Case Study

Select any one application mentioned below

Note: A report has to be submitted by every student at the end of the semester

1. Movie Review Classification
2. Text Classification
3. Multi label Classification
4. Email spam Filtering
5. Customer Segmentation
6. Anomaly Detection
7. Social Network Analysis
8. Product Recommendations (Frequently Bought Together)
9. Fraud Detection
10. Disease Diagnosis

Reference Books:

1. K.P. Soman, Shyam Diwakar and V. Ajay ―nIsight into Data mining Theory and Practice‖, Easter Economy Edition, Prentice Hall of India, 2006.
2. G. K. Gupta ―Itroduction to Data Mining with Case Studies‖, Easter Economy Edition, Prentice Hall of India, 2006.

Web Reference:

1. https://www.cs.waikato.ac.nz/ml/weka

MERN STACK DEVELOPMENT

III Year B. Tech. I semester

[Skill Course: IT]

Course Code: 24CT11SC02

L	T	P	C
1	0	2	2

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

CO1: **Illustrate** the fundamental concepts of JavaScript, TypeScript, and asynchronous programming with examples (L2).

CO2: **Identify** the components of MERN architecture, including MongoDB, Express.js, React.js, and Node.js, and their roles in web application development(L3).

CO3: **Demonstrate** CRUD operations using MongoDB and Mongoose for database management (L2).

CO4: **Build** RESTful APIs using Node.js and Express.js to enable server-side communication (L3).

CO5: **Develop** a full-stack web application with authentication and state management, integrating React.js frontend with a Node.js backend (L3).

Module-1: JavaScript Fundamentals

1. **Javascript Fundamentals:** Difference between Javascript and ECMAScript, Constants, Variables, Arrays, Objects, Functions, Arrow Functions, Modules, Spreading operators and destructiong assignment.
2. **Typescript:** Adding Typescript support to ECMAScript, Types, Interfaces.
3. **Asynchronous Programming**: Callbacks, Promises, Async/Await.

Module-2: Introduction to MEARN

1. Introduction to 3-Tier client/server architecture
2. Introduction to NoSQL
3. Introduction to REST API
4. Introduction to Web UI
5. Introduction to JSON
6. Overview of MERN Stack (MongoDB, Express.js, React, Node.js)

Module-3: MongoDB - NoSQL Database

1. Introduction to MongoDB and its features
2. Setting up MongoDB locally and on the cloud (MongoDB Atlas)Designing Collections and Documents

3. CRUD operations using MongoDB
4. Mongoose ODM for MongoDB schema and validation Aggregation framework and indexing

Module-4: Node.js - Server-Side Development

1. Setting up Node.js
2. Understanding Package Managers (npm, yarn)
3. Understanding package.json
4. Understanding Node.js Architecture
5. Create a Web Server in Node.js Restarting Node Application
6. File System & Module Management

Module-5: Express.js – Backend Framework

1. Express Development Environment Defining and Handling Routes Route and Query Parameters
2. Middleware (Custom, Built-in, Third-Party)
3. Connecting to MongoDB with Mongoose CRUD Operations
4. REST API Development
5. Testing API with Postman

Module-6: Angular.js – Frontend Development

1. Introduction to React.js & JSX
2. Environment Setup- installing Angular CLI
3. Create your first Angular app and understand its project structure
4. Work with components,directives,and pipes

Module-7:React.js -Frontend Development

1. Introduction to React.js & JSX
2. Functional Components
3. Nested Components
4. Properties and Property Drilling
5. Context Management with use Context
6. State Management using use State and use Effect
7. Global State Management with use Reducer

8. Routing with React Router
9. Styling with Tailwind

Module-8: Enterprise Application Development

1. Connecting the UI with Backend
2. Client requests using Fetch API
3. Client requests using Axios
4. Security and JWT
5. Preparing WebUI for distribution Deployment strategies

Case study:

Select any one practical application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

Building a Real-World MERN Stack Application with Authentication & CRUD Operations for the following problem statementsTech Stack:

1. **Frontend:** React.js, React Router, Axios
2. **Backend:** Node.js, Express.js, MongoDB, Mongoose.
3. **Authentication:** JWT (JSON Web Tokens), bcrypt (for password hashing).
4. **Database:** MongoDB (for storing users and tasks).

1. Healthcare Appointment Scheduling System:

A hospital faced challenges in managing patient appointments, leading to long wait times and inefficient resource utilization.

2. Food Delivery Service Expansion:

A food delivery startup struggled to scale operations to new cities due to logistical challenges.

3. Smart Agriculture System:

Farmers faced challenges in monitoring crop health and optimizing water usage.

4. Online Learning Platform for Kids:

Parents struggled to find engaging and educational content for their children during the pandemic.

5. Waste Management App:

A city faced challenges in managing waste collection and recycling efforts.

Text Books:

1. Full Stack JavaScript Development with MEAN, Colin J Ihrig, Adam Bretz, 1st edition, SitePoint, SitePoint
   Pty. Ltd., O'Reilly Media.
2. MongoDB – The Definitive Guide, 2nd Edition, Kristina Chodorow,O'Reilly.

Reference Books:

1. Robert W. Sebesta, "Programming with World Wide Web", Fourth Edition, Pearson, 2008.
2. Dayley B., "Node.js, MongoDB, and AngularJS Web Development", Addison Wesley Professional, 2014.
3. Vainikka J., "Full-Stack Web Development using Django REST Framework and React", 2018

Web References:

1. https://infyspringboard.onwingspan.com/en/app/toc/lex_17739732834840810000_shared/overview (HTML5)
2. https://infyspringboard.onwingspan.com/en/app/toc/lex_18109698366332810000_shared/overview (Javascript)
3. https://infyspringboard.onwingspan.com/en/app/toc/lex_32407835671946760000_shared/overview (Node.js & Express.js)
4. https://infyspringboard.onwingspan.com/en/app/toc/lex_9436233116512678000_shared/overview (Typescript)
5. https://infyspringboard.onwingspan.com/en/app/toc/lex_20858515543254600000_shared/overview (Angular JS)
6. https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_013177169294712832113_shared/overview (MongoDB)

DESIGN THINKING, INNOVATION & ENTREPRENEURSHIP

III Year B. Tech. I Semester
[Common to CSE, CSE (AI&ML), IT]

L	T	P	C
2	0	0	0

Course Code: 24IT11MC01

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

CO1: Outline a problem, apply methods of Empathy on user groups. [L2]

CO2: Describe and Define the problem specific to the user group. [L2]

CO3: Apply Ideation tools to generate Ideas to solve the problem. [L3]

CO4: Develop prototypes for the selected Ideas. [L3]

CO5 : Test the ideas and demonstrate Storytelling ability to present the Ideas. [L4]

Students shall form into groups and identify a problem (preferably societal problem with engineering orientation to solve) suitable for the design thinking and go through the process week-wise. At the end of each phase, brief documentation shall be submitted and a final report covering all phases has to be submitted at the end of the semester.

Weeks 1-3:

Introduction to Design Thinking: A primer on design thinking - Traditional approach, The new design thinking approach. Stages in Design Thinking: Empathize, Define, Ideate, Prototype, Test. Mindset for design thinking, Design thinking for product and process innovation, Difference between engineering design and design thinking.

Establishing Startups: Opportunity Scanning and Identification, Market Survey and assessment, Choice of technology and selection of business sites.

Case Studies: General, Engineering and Service applications.

Activities: Identify an Opportunity and Scope of the Project Explore the possibilities and Prepare design brief

Weeks 4-6:

Methods and Tools for Empathize and Define phases:

Empathize - Methods of Empathize Phase: Ask 5 Why / 5W+H questions, Stakeholder map, Empathy Map, Peer observation, Trend analysis

Define - Methods of Define Phase: Storytelling, Critical items diagram, Define success

Activities: Apply the methods of empathize and Define Phases Finalize the problem statement

Weeks 7-8:

Methods and Tools for Ideate phase:

Ideate - Brainstorming, 2X2 matrix, 6-3-5 method, NABC method;

Activities: Apply the methods of Ideate Phase: Generate lots of Ideas

Weeks 9-11:

Methods and Tools for Prototype Phase:

Prototype - Types of prototypes - Methods of prototyping - Focused experiments, Exploration map, Minimum Viable Product;

Activities: Apply the methods of Prototype Phase: Create prototypes for selected ideas

Weeks 12-13:

Methods and Tools for Test Phase:

Test - Methods of Testing: Feedback capture grid, A/B testing

Activities: Collect feedback; iterate and improve the ideas

Weeks 14-15:

Solution Overview - Create a Pitch - Plan for scaling up - Road map for implementation

Activities: Present your solution using Storytelling method

Week 16:

Project Submission: Fine tuning and submission of project report.

Reference Books:

1. Tim Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, HarperCollins e-books, 2009.
2. Michael Lewrick, Patrick Link, Larry Leifer, The Design Thinking Toolbox, John Wiley & Sons, 2020.
3. Michael Lewrick, Patrick Link, Larry Leifer, The Design Thinking Playbook, John Wiley & Sons, 2018.
4. Kristin Fontichiaro, Design Thinking, Cherry Lake Publishing, USA, 2015.
5. Walter Brenner, Falk Uebernickel, Design Thinking for Innovation - Research and Practice, Springer Series, 2016.
6. Gavin Ambrose, Paul Harris, Design Thinking, AVA Publishing, 2010.
7. Muhammad Mashhood Alam, Transforming an Idea into Business with Design Thinking, First Edition, Taylor and Francis Group, 2019.

8. S.Balaram, Thinking Design, Sage Publications, 2011.

Web References:

1. https://designthinking.ideo.com/
2. https://thinkibility.com/2018/12/01/engineering-vs-design-thinking/
3. https://www.coursera.org/learn/design-thinking-innovation
4. https://swayam.gov.in/nd1_noc20_mg38/preview

MACHINE LEARNING

III Year B. Tech. II Semester

[Common to CSE & IT]

L	T	P	C
3	0	0	3

Course Code: 24CT11RC23

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

CO1: Outline the fundamental concepts of machine learning. [L2]

CO2: Explain the life cycle of a machine learning project. [L2]

CO3: Build a variety of supervised learning models. [L3]

CO4: Apply the ensemble learning methods. [L3]

CO5: Experiment with the clustering techniques and dimensionality reduction models. [L3]

UNIT-I: **09 LECTURES**

Introduction: Relationship between artificial intelligence, machine learning, and deep learning. Main benefits of machine learning systems. Supervised, unsupervised and reinforcement learning, batch versus online learning, instance-based versus model-based learning. Main challenges of machine learning: irrelevant, insufficient, non- representative and poor quality data, problems of underfitting and overfitting models, testing and validating ML models.

UNIT-II: **09 LECTURES**

End-to-End Machine Learning Project: Problem formulation: working with real data, looking at the big picture, framing the problem and select a performance measure. Expolatory data analysis: data collection, cleaning, visualization and preparation. Model building: selection and training a model, evaluating and fine-tuning the model through cross-validation, grid_search_CV, and randomized_search_CV. Launching an ML system: evaluate an ML system on the test set, launching, monitoring, and maintaining an ML system.

UNIT III: **12 LECTURES**

Supervised Learning: Binary, multiclass and multilabel classification problems. Performance measures. Linear regression, gradient descent, polynomial regression, regularization, logistic regression and softmax regression. Popular classification models: k-nearest-neighbours algorithm (k-NN), naïve Bayes classifier, and decision trees.

UNIT-IV: **10 LECTURES**

Advanced Supervised Learning: Linear Discriminant Analysis, Support Vector Machines: linear and nonlinear SVM classification, SVM regression. Ensemble Learning and Random Forests: introduction,

voting classifiers, bagging and pasting, random forests, boosting and stacking.

UNIT-V: **10 LECTURES**

Dimensionality Reduction: The curse of dimensionality, main approaches for dimensionality reduction, principal component analysis, incremental, randomized, and kernel PCA algorithms. **Introduction to Artificial Neural Networks:** Backpropagation algorithm, solving classification and regression problems using ANNs.

Text Books:

1. Hands-On Machine Learning with Scikit-Learn, Keras, andTensorFlow,‖ 3rd Edition, Aurélien Géron (2022), O,,Reilly Publications.
2. Machine Learning, Tom M. Mitchell (1998), McGraw-Hill.

Reference Books:

1. Python Machine Learning By Example, Yuxi Liu (2024), Packt Publishing.

Web References:

1. https://onlinecourses.nptel.ac.in/noc25_cs46/preview
2. https://www.coursera.org/specializations/machine-learning-introduction

CRYPTOGRAPHY AND NETWORK SECURITY

III Year B. Tech. II semester

[Common to CSE &IT]

Course Code: 24CI11RC02

L	T	P	C
3	0	0	3

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

CO1: Outline security attacks, services, mechanisms, network security models, and classical encryption techniques. [L2]

CO2: Utilize symmetric key cryptography principles, including modern block ciphers, DES, AES, Blowfish, IDEA, and block cipher operations, for secure communication**.** [L3]

CO3: Make use of public key cryptography, number theory, cryptographic hash functions, and digital signatures for secure communication and authentication. [L3]

CO4: Demonstrate key management, compare key distribution methods, and evaluate security protocols for secure communication. [L2]

CO5: Apply transport-level security (TLS, SSL) and system security measures, including IDS, firewalls, and malicious program protection for secure communication. [L3]

UNIT-I **08 Lectures**

Computer and network security concepts: Security Attacks, Security Services & Mechanisms, A model for network security, Classical Encryption Techniques – Substitution techniques, Transposition techniques.

UNIT-II **12 Lectures**

Symmetric Key Cryptography: Introduction to Modern Symmetric Key ciphers, Modern Block Ciphers, Components of Modern Block Ciphers, Traditional Block Cipher structure, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Blowfish, IDEA, and Block Cipher Operations.

UNIT-III **12 Lectures**

Public Key Cryptography: Principles of public key Cryptography, Mathematics of Asymmetric key cryptography-Introduction to Number Theory-Prime numbers and Relatively Prime Numbers, Modular Arithmetic, Fermat‘s and Euler‘s Theorems, The Chinese Remainder Theorem, Discrete Logarithms, The RSA Algorithm, Diffie Hellman key exchange.

Cryptographic Hash Functions- Application of Cryptographic Hash functions, Requirements & Security, SHA, Message Authentication Codes, Digital Signatures.

UNIT-IV **10 Lectures**

Key Management: Symmetric Key Distribution, Kerberos, Public Key Distribution.

E-mail & IP Security: Internet Mail architecture, E-mail security using PGP, IPSec-IP Security-Overview, IP Security policy, Encapsulating Security Payload (ESP), Authentication Header (AH).

UNIT-V **08 Lectures**

Transport Level Security: Web Security considerations, Transport Layer Security (TLS), Secure Socket Layer (SSL).

System Security: Malicious Programs, IDS, Firewalls: Characteristics, Types of Firewalls.

Textbooks:

1. Cryptography and Network Security- Principles and Practise by William Stallings , Seventh Edition
2. Cryptography and Network Security- Behrouz A Forouzan, Debdeep Mukhopadhyay, McGrawHill, 3rd Edition.2015.

Reference Books:

1. Cryptography and Network Security- by Atul Kahate

Web References:

1. https://onlinecourses.nptel.ac.in/noc25_cs16/preview

OBJECT ORIENTED SOFTWARE ENGINEERING

III Year B. Tech. II Semester

[Common to CSE & IT]

Course Code: 24CI11RC03

L	T	P	C
3	0	0	3

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

CO1: Outline the basics of Software Engineering, its process models and the idea of Requirements Engineering. (L2)

CO2: Demonstrate the usage different UML diagrams with a user centered approach. (L2)

CO3: Make use of the software architecture and design patterns that are studied. (L3)

CO4: Apply the concepts of Software Testing to Real-time Products to enhance Software Quality. (L3)

CO5: Examine the Process management and cost estimation concepts to make a Software Project Successful. (L4)

UNIT-I **12 Lectures**

Introduction to Software Engineering: The Nature of Software, The Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Technology.

Agility and Requirements: Agility, Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, A Tool Set for the Agile Process, Software Engineering Knowledge, Core Principles. Requirements Engineering: Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements, Validating Requirements.

UNIT-II **10 Lectures**

Unified Modelling Language & Use Case Modelling: Introduction to UML, Modelling Concepts, Types of UML Diagrams with Examples; User-Centred Design, Characteristics of Users, Developing Use-Case Models of Systems, Use-Case Diagram, Use-Case Descriptions, Basics of User Interface Design, Usability Principles, User Interfaces.

Class Design and Class Diagrams: Essentials of UML Class Diagrams, Associations and Multiplicity, Other Relationships, Generalization, Instance Diagrams, Advanced Features of Class Diagrams, Interaction and Behavioural Diagrams: Interaction Diagrams, State Diagrams, Activity Diagrams, Component and Deployment Diagrams.

UNIT-III **10 Lectures**

Software Design and Architecture: Process of Design, Principles Leading to Good Design, Techniques for Making Good Design Decisions, Good Design Document; Pattern Introduction.

Design Patterns: Abstraction-Occurrence Pattern, General Hierarchical Pattern, Play-Role Pattern, Singleton Pattern, Observer Pattern, Delegation Pattern, Adaptor Pattern, Façade Pattern, Immutable Pattern, Read-Only Interface Pattern and The Proxy Pattern; Software Architecture Contents of Architecture Model, Architectural Patterns: Multilayer, Client-Server, Broker, Transaction Processing, Pipe & Filter and MVC Architectural Patterns.

UNIT-IV **10 Lectures**

Software Testing: Overview of Testing, Testing Concepts, Testing Activities, Testing Strategies, Unit Testing, Integration Testing, Function Testing, Structural Testing, Class Based Testing Strategies, Use Case/Scenario Based Testing, Regression Testing, Performance Testing, System Testing, Acceptance Testing, Installation Testing, O.O. Test Design Issues, Test Case Design, Quality Assurance, Root Cause Analysis, Post-Mortem Analysis.

UNIT-V **08 Lectures**

Software Process Management: Introduction to Software Project Management, Rationale Management, Configuration Management, Activities of Software Project Management, Structure of Project Plan, Software Engineering Teams, Software Cost Estimation, Project Scheduling, Tracking and Monitoring.

Textbooks:

1. Software Engineering: A Practitioner's Approach, Roger S Pressman.
2. Object-Oriented Software Engineering Practical software development using UML and Java by Timothy C. Leth bridge & Robert Langaniere, McGraw-Hill

Reference Books:

1. A Practical Guide to Testing Object-Oriented Software, John D. McGregor; David A. Sykes, Addison-Wesley Professional.
2. Object-Oriented Software Engineering: Using UML, Patterns and Java, Bernd Bruegge and Allen H. Dutoit, 2nd Edition, Pearson Education Asia.

Web References (e-Resources):

1. Udemy: Object-Oriented Software Engineering with Java; https://tinyurl.com/bdfy26z9
2. NPTEL: Object oriented analysis and design; https://onlinecourses.nptel.ac.in/noc19cs48/preview

MACHINE LEARNING LAB

III Year B. Tech. II semester
[Common to CSE & IT]

Course Code: 24CT11RC25

L	T	P	C
0	0	3	1.5

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

CO1: Apply exploratory data analysis and categorical encoding techniques to understand the data and make it ready for model building. (L3)

CO2: Build ML models based on classic algorithms like kNN, linear regression and logistic regression. (L3)

CO3: Develop SVM model from scratch. (L3)

CO4: Construct ML models based on ensemble techniques. (L3)

CO5: Make use of PCA, K-Means and EM to construct unsupervised learning models. (L3)

Module-1: Exploratory Data Analysis

Illustrate Exploratory Data Analysis using Pandas or Matplotlib: Getting a quick statistical summary of a dataset, Checking Missing Values, Removing Duplicates, Showing Correlations, and Looking for Outliers using BoxPlots.

Module-2: Categorical Encoding

Write a program to implement Categorical Encoding: One-hot/Label/Ordinal Encodings.

Module-3: Decision Trees

Use sklearn to build a decision tree for IRIS data set and visualize the decision tree. Use the decision tree built to classify a new data point using both predict and predict_proba methods. Need to demonstrate your understanding of sklearn's decision tree classifier by using both entropy and Gini index.

Module-4: k-Nearest Neighbors algorithm

Use sklearn's k-Nearest Neighbors algorithm to classify the iris dataset. Print both correct and wrong predictions. You need to print the attributes values for wrong predictions.

Module-5: Linear Regression

Use linear regression algorithm to predict median price of a district by training a regression model on the Boston housing dataset.

Module-6: Logistic Regression

Use logistic regression algorithm to classify a handwritten digit as even or odd by training a logistic regression model on the MNIST dataset.

Module-7: Support Vector Machines

Implement, visualize and test Support Vector Machines in Python (not allowed to use sklearn's Support Vector classes).

Module-8: Random Forest algorithm

Run decision tree classification and random forest algorithms on a dataset and confirm that ensembling indeed improves performance significantly reducing the variance component (using bias- variance decomposition). Also, print feature importances as found by the random forest algorithm.

Module-9: Gradient Boosting algorithm

Use sklearns, GradientBoostingRegressor and demonstrate the use of early stopping to find the optimal number of trees in the GBRT ensemble.

Module-10: Principle Component Analysis

Implement and test Principle Component Analysis in Python (not allowed to use sklearn's PCA classes).

Module-11: Clustering algorithms

Apply expectation-maximization (EM) algorithm to cluster a Heart Disease Data Set. Use the same data set for clustering using the k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering.

Case Study: Select any practical application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

1. Customer Segmentation for a Retail Business Concepts: Clustering (k-Means, EM), PCA

Description: Analyze customer purchase history and segment customers for personalized marketing strategies.

2. Credit Card Fraud Detection

Concepts: Logistic Regression, Random Forest, Decision Trees, Gradient Boosting Description: Use transaction data to predict fraudulent transactions, improving fraud detection mechanisms.

3. Employee Attrition Prediction

Concepts: Logistic Regression, Decision Trees, Categorical Encoding Description: Identify key factors influencing employee retention in a company.

4. Predicting House Prices

Concepts: Linear Regression, PCA, Gradient Boosting

Description: Build a model to predict house prices based on historical housing data.

5. Fake News Detection

Concepts: NLP, Logistic Regression, Decision Trees, SVM

Description: Classify news articles as real or fake using text-based features.

6. Movie Recommendation System Concepts: k-NN, PCA

Description: Build a recommendation system based on user preferences and ratings.

7. Stock Market Price Prediction

Concepts: Linear Regression, Gradient Boosting

Description: Predict stock prices based on historical trends and market indicators.

8. Healthcare Readmission Prediction Concepts:

Logistic Regression, Decision Trees

Description: Predict which patients are at high risk of hospital readmission.

9. Air Quality Prediction

Concepts: Linear Regression, PCA, Decision Trees

Description: Predict air pollution levels based on meteorological and traffic data.

10. Predicting Loan Default Risk

Concepts: Logistic Regression, Random Forest

Description: Assess whether a loan applicant is likely to default.

11. E-commerce Product Review Sentiment Analysis

Concepts: NLP, SVM, k-NN

Description: Classify product reviews into positive, neutral, or negative sentiments.

12. Energy Consumption Forecasting

Concepts: Linear Regression, Gradient Boosting

13. Description: Predict energy consumption trends for efficient energy planning.

    1. Traffic Accident Severity Prediction
    2. Concepts: Decision Trees, k-NN, PCA

14. Description: Predict accident severity based on road conditions, weather, and driver behavior.

    1. Disease Diagnosis from Symptoms Concepts:
    2. Decision Trees, Random Forest, SVM

15. Description: Predict diseases based on patient symptoms.
    1. Customer Churn Prediction for Telecom
    2. Concepts: Logistic Regression, Decision Trees, Categorical Encoding Description: Predict which customers are likely to switch telecom providers.

Reference Books:

1. Aurélien Géron (2022), ―Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow,‖ 3rd Edition, O,,Reilly Publications.
2. Yuxi Liu (2024), ―Python Machine Learning By Example,‖ Packt Publishing.

Web References (e-Resources):

1. https://onlinecourses.nptel.ac.in/noc25_cs46/preview
2. https://www.coursera.org/specializations/machine-learning-introduction

CRYPTOGRAPHY AND NETWORK SECURITY LAB

III Year B. Tech. II semester

[Common to CSE & IT]

Course Code: 24CI11RC04

L	T	P	C
0	0	3	1.5

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

CO1: Demonstrate encryption techniques using XOR and bitwise operations in C, and develop Caesar, Substitution, and Hill cipher encryption techniques in Java for secure communication. (L2)

CO2: Experiment with symmetric-key encryption using DES, Blowfish, Rijndael-AES, Algorithms for secure data transmission. (L3)

CO3: Develop public-key encryption techniques like RSA algorithm and the Diffie-Hellman Key Exchange in Java for secure communication. (L3)

CO4: Apply cryptographic Hash functions by implementing SHA-512 and MD5 in Java for secure message digest generation. (L3)

CO5: Analyse network security practices using diagnostic tools, packet capture, intrusion detection, and ARP poisoning to identify and mitigate potential threats. (L4)

Implement the programs using C/Java

Module-1 : Basic Encryption Techniques

1. Write a program that contains a string (char pointer) with a value= ―Hello world‖. The Program
   Should XOR each character in this string with 0 and displays the encrypted result.
2. Write a java program that contains a string (char pointer) with a value- Hello world!. The Program should AND or and XOR each character in this string with 127 and display the result.

Module-2: Classical Encryption Techniques

1. Write a program to perform encryption and decryption using the following algorithms
   a. Substitution cipher b. Hill Cipher c. Play fair Cipher

Module-3: Symmetric Key Cryptographic Encryption Algorithms

1. Write a program to implement the DES algorithm logic
2. Write a program to implement the AES algorithm

Module-4 : Asymmetric Key Cryptographic Encryption Algorithms

1. Write a program to implement RSA algorithm.

Module-5: Key-Exchange-technique

1. Implement the Diffie-Hellman Key Exchange mechanism.

Module-6: Data Integrity with Cryptographic Hash Functions

1. Calculate the message digest of a text using the MD5 algorithm.
2. Calculate the message digest of a text using the SHA-512 algorithm.

Module-7: **Network Security Basics:**

1. Find the IP address & MAC address of your machine.
2. Find the neighbouring machines in your network.
3. Check if a server is up and running.

Module-8: Tools for Diagnosis Network security

1. Run Tcpdump/windump utility with atleast 4 options.
2. Capture the packets in your system using Wireshark and analyse any one TCP packet in detail.

Module-9: Tools for Intrusion Detection in Network

1. Use snort to detect intrusion packets.
2. Demonstrate ARP Poisoning.

CASE STUDY

Select any one application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

1. Implement Brute force Technique to guess the plaintext from ciphertext.
2. Design a simple encrypted chat application that enables secure communication between Alice,Bob.
3. Secure File Transfer System: A secure system where users upload files, which are encrypted using AES.
4. Design Secure Online Banking System (HTML, JS, Java) –where Encrypt transactions

with AES, RSA for authentication, Diffie-Hellman for secure key exchange.

5. Design Secure Web Login with Two-Factor Authentication (2FA): A secure login system where User passwords are stored securely using SHA-256 hashing, and login requires a one-time password (OTP) sent via email or SMS.
6. Design Cryptographic Text Edito (C/Java) – Encrypt text files with Caesar, Substitution, or Hill Cipher, decrypt on demand.
7. Design Secure QR Code Generator (Java) – AES/RSA encryption for text, convert to QR code, scan to decrypt.
8. Design a Time-based One-Time Password (TOTP) generates an OTP that changes every 30 seconds.

Reference Books:

1. Computer Security - Principles and Practices, 2ndEdition by William Stallings, Pearson Education, Inc.
2. Cryptography and Network Security, 3rd Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill,2015

OBJECT ORIENTED SOFTWARE ENGINEERING LAB

III Year B. Tech. II semester

[Common to CSE & IT]

Course Code: 24CI11RC05

L	T	P	C
0	0	3	1.5

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

CO1: Understand the Requirements & Prepare Documentation. (L2)

CO2: Make use of various design diagrams for the gathered requirements. (L3)

CO3: Construct the UML Diagrams. (L3)

CO4: Analyse the quality in object-oriented software engineering. (L4)

CO5: Understand the testing problems and test whether all the requirements specified have been achieved or not. (L2)

Module-1: Documentation including

1. A problem statement
2. E-R Diagram

Module-2: Requirements document

1. A Requirements Analysis Document.
2. A System Requirements Specification**.**
3. A Software Requirements Specification.

Module-3: Design document

1. A Software Design Description and a System Design Document.
2. System Design Document Template.

Module-4: UML Diagrams (Structural)

1. Introduction to Class Diagram.
2. Introduction to Object Diagram.
3. Introduction to component Diagram.
4. Introduction to Deployment Diagram.

Module-5: UML Diagrams (Behavioural)

1. Introduction to Use Case Diagram.
2. Introduction to Sequence Diagram.
3. Introduction to Activity Diagram.

4. Introduction to State Diagram.

Module-6: A Test Specification

1 Test Id
2 Test Name
3 Test Type
4 Expected Result
5 Actual Result
6 Pass/Fail

Module-7: Manuals/guides for

a. Users and associated help frames
b. Programmers
c. Administrators (installation instructions)

Module-8: A project plan and schedule setting out milestones, resource usage and estimated costs.

i. Parametric models.
ii. Cost Estimations.

Module-9: A quality plan setting out quality assurance procedures.

1. Software Quality Assurance (SQA) with all types of testing like functional testing, unit testing, integration testing, system testing, user acceptance testing etc.

CASE STUDY

Select any one application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, and output through UML diagrams.

List of Case Studies:

1. Hotel Management System,
2. Database Manager for Magazine Agency or Newspaper Agency,

3. Ticket Booking for Performances,
4. Preparing Greeting Cards & Birthday Cards
5. Personal Accounts - Insurance, Loans, Mortgage Payments, Etc.,
6. Doctor's Diary & Billing System
7. Personal Bank Account
8. Class Marks Management
9. Hostel Accounting
10. Video Tape Library,
11. History of Cricket Scores,
12. Cable TV Transmission Program Manager,
13. Personal Library.
14. Sailors Database.
15. Suppliers and Parts Database.
16. Travel Guide.
17 Online Shopping.
18. Online Voting System.
19. Airline Reservation System.
20. Online Flower Bookings.

References:

1. Project-based software engineering: An Object-oriented approach, Evelyn Stiller, Cathie LeBlanc, Pearson Education
2. Visual Modelling with Rational Rose 2002 and UML, Terry Quatrini, Pearson Education 3. UML2 Toolkit, Hans -Erik Eriksson, etc; Wiley

SOFT SKILLS

III Year B. Tech. II semester

[Skill Course: Common to CSE & IT]

Course Code: 24HE11SC01

L	T	P	C
1	0	2	2

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

CO1: **Apply** effective communication skills in personal and professional settings. (L3)

CO2: **Develop** and implement effective goal setting and time management strategies. (L3)

CO3: **Function** as a leader and a member of a team. (L4)

CO4: **Analyze** data to write concise technical reports (L4), **Organize** Effective meetings and record minutes of the meeting. (L3)

CO5: **Take part** in group discussions (L4), **Build** resumes to successfully navigate job interviews. (L3)

UNIT-I

Communication skills: Verbal & Non-verbal communication- Personal grooming (Etiquette, Attitude, Body Language), Posture, Gestures, Facial Expressions, Eye contact, Space Distancing, Interpersonal communication- Presentation skills, Public speaking, - Cross Cultural communication.

UNIT-II

Goal Setting and Time Management: Immediate, Short term, Long term, SMART Goals, Strategies to Achieve goals, Time Management Skills, Identifying Time Wasters, Stress Busters.

UNIT-III

Leadership and Team Management: Qualities of a Good Leader, Team Dynamics, Leadership Styles, Decision Making, Problem Solving, Negotiation Skills.

UNIT - IV

Business communication: Types of reports- Technical report writing, Proposals, SOP; Planning

for effective meetings, Minutes of the Meetings.

UNIT-V

Interview Skills: Group Discussions, Resume preparation and Mock interviews.

Reference Books:

1. Krannich, Caryl, and Krannich, Ronald L. Nail the Resume! Great Tips for Creating Dynamite Resumes. United States, Impact Publications, 2005.
2. Hasson, Gill. Brilliant Communication Skills. Great Britain: Pearson Education, 2012
3. Prasad, H. M. How to Prepare for Group Discussion and Interview. New Delhi: Tata McGraw-Hill Education, 2001.
4. Pease, Allan. Body Language. Delhi: Sudha Publications, 1998.
5. Rizvi, Ashraf M. Effective Technical Communication: India, McGraw-Hill Education. 2010
6. Thorpe, Edgar &Showick Thorpe. Winning at Interviews. 2nd Edition. Delhi: Dorling Kindersley, 2006.
7. N.L.Gupta. Cross cultural Communication: Global perspectives. Concept Publishing Company, 1998.
8. Peter Hartley and Clive G. Bruckmann. Business communication. 2002.

Web References:

1. https://www.hbr.org
2. A step-by-step guide to writing a technical report | Indeed.com UK
3. https://www.grammarly.com/blog/business-writing/meeting-minutes/
4. http://www.youtube.com/@PebblesLanguageLearning
5. https://www.onlinecourses.nptel.ac.in/noc20_hs60/preview

IPR & PATENTS

III Year B. Tech. II Semester
[Common to CSE, CSE (AI&ML), IT]

L	T	P	C
2	0	0	0

Course Code: 24HM11MC02

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

CO1: Understand basic concepts of Intellectual property rights, IPR tool kit and its importance in the global scenario. (L2)

CO2: Apply the knowledge on patents for registration process and understand recent developments in patent system. (L3)

CO3: Demonstrate an understanding about copyright protection, the registration process and legal remedies available in case of infringement (L2)

CO4: Integrate and understand the concept of Trademarks and industrial design, their registration, infringement and related laws (L3)

CO5: Identify principles of trade secrets, Semi conductor Integrated Circuits Layout Design, Geographical Indications, Protection of Plant Varieties & Farmers' right and emerging areas of IP (L4)

Unit-I **6 Lectures**

Introduction and Evolution of IP system : Concept of property, Intellectual Property Rights (IPR), Importance of IP, Value creation through IP, Advantages of IP protection, Competitive advantage, History and rationale behind development of IP system, WTO and TRIPS agreement, Role of WIPO. Major forms of IP in India and globally.

TextBook-1: Chapter 1, 2, 3, 11 & 12, Text Book -2: Chapter1, 2 3 & 4 Text Book -4: Chapter 41

Unit-II **6 Lectures**

Patent, Patent filing and prosecution : Concept, Life of patent, Rights of Patentee, Criteria of patentability- novelty, non- obviousness, and utility, Non-patentable inventions. Prior art search, Process of obtaining a patent in India, Convention application, Patent Cooperation Treaty (PCT), Patent Infringement, IP commercialization: Licensing & Royalty; Technology Transfer, Compulsory License, Pat informatics. Case studies on patenting of life saving drugs and their implications. TextBook-1: Chapter 3 & 4, Text Book -2: Chapter 4,5,6 &10

Unit-III **6 Lectures**

Trademarks and Industrial Design : Types of trademarks, Trademark Registration process, Trademark Infringement, Post registration procedures & Trade Mark maintenance, Genericide Concept of Industrial design, Design registration, Design infringement. TextBook-1: Chapter 7, 8 & 10

Unit-IV: **6 Lectures**

Copyrights and related rights : Subject Matters of Copyright, Copyright registration, Copyright infringement, Section 52 of Indian Copyright Act– Rights Afforded by Copyright Law, Fair use –Right to Prepare Derivative Works - Plagiarism vs Copyright infringement- Copyright pertaining to Software/Internet and other Digital media. TextBook-1: Chapter 5 & 6

Unit-V: **6 Lectures**

Other forms of IP and Emerging areas of IP :Trade Secret- Maintaining Trade Secret –Employee Confidentiality Agreement – Semiconductor Integrated Circuits Layout Design, Geographical Indications, Protection of Plant Varieties & Farmers' rights, Traditional knowledge- **Case studies on biopiracy.** IP in bank loan, and insurance, Use of artificial intelligence in IP enforcement. TextBook-1: Chapter 9, 10 Text Book -2 Chapter 11

Text Books

1. Kompal Bansal &Parishit Bansal "Fundamentals of IPR for Engineers", BS Publications (Press)
2. Prabhuddha Ganguli: ' Intellectual Property Rights" Tata Mc-Graw Hill, New Delhi
3. Deborah E.Bouchoux: "Intellectual Property". Cengage learning , New Delhi
4. M. Ashok Kumar and Mohd. Iqbal Ali: "Intellectual Property Right" Serials Pub.

Reference Materials

1. The Indian Patents Act 1970 (as amended in 2005)
2. The Indian Copyright Act 1950 (as amended in 2017)
3. Indian Trademarks Act 1999
4. The Indian Industrial Designs Act 2000
5. The Protection of Plant Varieties and Farmers' Right Act 2001
6. R. Radha Krishnan, S. Balasubramanian: "Intellectual Property Rights", Excel Books. New Delhi.
7. Geographical Indications of Goods Act 1990

Web References (e-Resources):

1. https://www.wipo.int/en/web/wipo-academy/
2. Inventing the Future: An Introduction to Patents for small and medium sized enterprises, WIPO publication No 917 www.wipo.int/ebookshop
3. Looking Good: An Introduction to Industrial Designs for Small and Medium sized Enterprises; WIPO publication No.498 www.wipo.int/ebookshopGanguli Prabuddha
4. "Geographical Indications-its evolving contours accessible in http ips. nminsoda/files/2012/05/ main book pdf (2009)

DEEP LEARNING ESSENTIALS WITH PYTHON

IV Year B. Tech. I semester

[Skill Course: Common to CSE & IT]

Course Code: 24CI11SC01

L	T	P	C
1	0	2	2

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

CO1: Implement deep neural networks to solve real world problems. (L3)

CO2: Apply convolution neural networks to solve computer vision problems. (L3)

CO3: Apply recurrent neural networks to solve time series problems. (L3)

CO4: Choose appropriate pre-trained model to solve real time problems. (L3)

CO5: Implement generative AI models for real world applications. (L3)

Module-1: Multilayer Perceptron MLP

Implement multilayer perceptron algorithm for MNIST Hand written Digit Classification.

Module-2: Binary Classification

Design a neural network for classifying movie reviews using IMDB dataset.

Module-3: Multiclass Classification

Design a neural Network for classifying newswire articles using Reuters dataset.

Module-4: Regression algorithm

Design a neural network for predicting house prices using Boston Housing Price dataset.

Module-5: Convolution, Pooling and Strides

Build a Convolution Neural Network for MNIST Hand written Digit Classification. Try with both convolution and pooling layers, with just convolution layers (possibly with stride greater than 1).

Module-6: Data Augmentation

Build a Convolution Neural Network for simple image (dogs and Cats) classification with and without data augmentation.

Module-7: Recurrent Neural Networks

Implement a deep Recurrent Neural Network for a time series analysis.

Module-8: LSTM and GRU Networks

Implement a deep LSTM and GRU networks for a time series analysis.

Module-9: Transfer Learning

Use a pre-trained convolution neural network (VGG16) for image classification.

Module-10: Autoencoders

Implement dimensionality reduction using an autoencoder.

Module-11: GANs

Implement GAN to generate fashion item images.

Case Study: Select any practical application mentioned below.

Note: A report has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

1. Handwritten Signature Verification

Concepts: MLP, CNN, Transfer Learning Description: Develop a deep learning model to verify handwritten signatures and detect forgeries.

2. Skin Disease Classification

Concepts: CNN, Transfer Learning, Data Augmentation

Description: Use a CNN-based model to classify different types of skin diseases from medical images.

3. Real-Time Face Mask Detection

Concepts: CNN, Transfer Learning (VGG16), Data Augmentation

Description: Build a model that detects whether a person is wearing a mask or not in real-time.

4. Sentiment Analysis of Social Media Posts Concepts: Multiclass Classification, RNN, LSTM

Description: Classify tweets or social media posts into positive, negative, or neutral sentiments.

5. Fake News Detection

Concepts: Binary Classification, LSTM, GRU

Description: Build a model to classify news articles as real or fake.

6. Weather Forecasting using Time-Series Data Concepts: RNN, LSTM, GRU Description: Predict weather conditions (e.g., temperature, humidity) using historical data.

7. AI-Powered Chatbot for Customer Support Concepts: NLP, LSTM, GRU Description: Develop a chatbot that can respond to customer queries using deep learning.

8. Music Genre Classification Concepts: CNN, RNN, Transfer Learning

Description: Classify songs into different genres based on their audio features.

9. Hand Gesture Recognition Concepts: CNN, Data Augmentation

Description: Build a model to recognize hand gestures for controlling smart devices.

10. Autonomous Vehicle Lane Detection Concepts: CNN, Transfer Learning

Description: Develop a lane detection system to assist in self-driving cars.

11. Super-Resolution Image Enhancement Concepts: Autoencoders, GANs

Description: Implement a deep learning model to improve image resolution.

12. Fashion Item Generation using GANs

Concepts: GANs Description: Train a Generative Adversarial Network (GAN) to create new fashion designs.

13. Personalized Movie Recommendation System Concepts: Neural Networks, Autoencoders

Description: Build a model that recommends movies based on user preferences.

14. 3D Object Reconstruction from 2D Images Concepts: CNN, Autoencoders

Description: Develop a deep learning model to reconstruct 3D objects from 2D images.

15. Autonomous Speech-to-Text Conversion Concepts: RNN, LSTM, GRU Description: Build a deep learning model that converts spoken language into text with high accuracy.

Reference Books:

1. "Deep Learning with Python," 2nd Edition, Francois Chollet (2021), Manning Publications.
2. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," 3rd Edition, Aurélien Géron (2022), O'Reilly Publications.

Web References (e-Resources):

1. https://onlinecourses.nptel.ac.in/noc20_cs62/preview
2. https://www.coursera.org/specializations/deep-learning

API AND MICRO SERVICES

[Professional Elective]

Course Code: 24CT11PE01

L	T	P	C
3	0	0	3

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

CO1: Identify the core components of the Spring Framework, including IoC containers, dependency injection techniques, and module configurations. (L3)

CO2: Build Spring Boot applications with autowiring, logging, and Aspect-Oriented Programming (AOP) to enhance modularity and maintainability. (L3)

CO3: Utilize Spring Data JPA for database interactions, implementing CRUD operations, pagination, query methods, and transaction management. (L3)

CO4: Develop SOAP-based and RESTful web services, applying SOA principles and best practices for service implementation. (L3)

CO5: Make use of Spring REST features such as request handling, response processing, API versioning, exception handling, and security mechanisms. (L3)

UNIT-I **10 Lectures**

Spring 5 Basics: Introduction to Spring Framework, Spring Framework - Modules, Configuring IoC container using Java-based configuration, Introduction to Dependency Injection, Constructor Injection, Setter Injection, AutoScanning.

UNIT-II **10 Lectures**

Spring Boot: Creating a Spring Boot Application, Spring Boot Application Annotation, Autowiring, Scope of a bean, Logger, Introduction to Spring AOP, Implementing AOP advices, Best Practices : Spring Boot Application.

UNIT-III **12 Lectures**

Spring Data JPA with Boot: Limitations of JDBC API, Need for Spring Data JPA, Spring Data JPA with Spring Boot, Spring Data JPA Configuration, Pagination and Sorting, Query Approaches, Named Queries and Query, Overview of Spring Transaction, Spring Declarative Transaction, Update Operation in Spring Data JPA, Custom Repository Implementation, Best Practices - Spring Data JPA.

UNIT-IV **08 Lectures**

Web Services*:* Introduction to Web services, SOA - Service Oriented Architecture, Types of Web Services, SOAP based Web Services, RESTful Web Services, How to creation of RESTful.

UNIT-V **10 Lectures**

Spring REST: Spring REST - An Introduction, Creating a Spring REST Controller, @RequestBody and ResponseEntity, Parameter Injection, Usage of @PathVariable, @RequestParam and @MatrixVariable, Exception Handling, Data Validation, Creating a REST Client, Versioning a Spring REST endpoint, Enabling CORS in Spring REST, Securing Spring REST endpoints.

Text Books:

1. "Spring in Action," Craig Walls, 5th Edition, Manning Books, 2018. ISBN: 978- 1617294945.
2. "Pro Spring Boot 2: An Authoritative Guide to Building Microservices, Web and Enterprise Applications, and Best Practices," Felipe Gutierrez, Apress, 2018. ISBN: 978- 1484236758.

Reference Books:

1. "Mastering Spring Boot 3.0**,**" Ranga Rao Karanam, Packt Publishing, 2022.
2. "Mastering Spring Boot 2," Dinesh Rajput, P*ackt Publishing, 2019.*

Web References (e-Resources):

1. Infosys Spring board courses: https://infyspringboard.onwingspan.com/web/en/app/toc/lex_19872203945352840000_shared/overview [Spring 5 Basics]
2. https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01296689056211763272_shared/overview [Spring 5 Basics with Spring Boot]
3. https://infyspringboard.onwingspan.com/en/app/toc/lex_4313461831752789500_shared/overview [Spring Data JPA with Boot]
4. https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01273190096390553 6190_shared/overview [Spring REST]
5. https://docs.spring.io/spring-framework/reference/
6. https://docs.spring.io/spring-boot/
7. https://docs.spring.io/spring-data/jpa/reference/
8. https://spring.io/guides/gs/rest-service
9. https://spring.io/guides/gs/securing-web

BLOCKCHAIN TECHNOLOGY

[Professional Elective]

Course Code: 24CI11PE01

L	T	P	C
3	0	0	3

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

CO1: Summarize the Blockchain ecosystem, properties, advantages and limitations. (L2)

CO2: Explain Blockchain's decentralized and distributed nature. (L2)

CO3: Interpret the components and transaction flow of Hyperledger Fabric within private blockchain ecosystems. (L2)

CO4: Apply Intelligent techniques in Blockchain. (L3)

CO5: Utilize advanced blockchain concepts to design and implement various blockchain applications. (L3)

UNIT-I **10 Lectures**

Blockchain and its evolution: Introduction to Blockchain, Scenarios, Characteristics of Blockchain, History of Blockchain, Stages in Blockchain Evolution, Restrictions on Sharing Ledgers, Block structure, Chaining of Blocks, Security on Blockchain, Type of players in Blockchain ecosystem.

UNIT-II **12 Lectures**

Distributed Consensus: Decentralized Blockchain, Merkle Tree, Consensus: Byzantine General Problem, Real Need for Mining, Proof of Work (PoW), Proof of Stake (PoS), Proof of Burn (PoW), Proof of Concept (PoC), Practical Byzantine Fault Tolerance (EPBFT).

Bitcoin: Bitcoin, Transactions, Bitcoin properties - Transaction life cycle - creation of coin -sending payments - double spending using blockchain - bitcoin anonymity, Bitcoin limitations.

UNIT-III **10 Lectures**

Ethereum and Smart Contracts: Ethereum ecosystem, Ethereum development, Ethereum Tool Stack, Ethereum virtual machine, How Mining Works, Ethereum Workflow, Smart Contract programming, Integrated Development Environment, Ganache, Ethereum Accounts, My Ether Wallet, Ether Scan, Decentralized application, Ethereum Networks/ Environments, Metamask, Tuna Fish Use case Implementation.

UNIT-IV **9 Lectures**

Private Blockchain with Hyperledger: Introduction, Private Blockchain Ecosystem, Hyperledger Fabric Transaction Flow, FabCar Use case Implementation, Invoking Chaincode functions using Client Application.

UNIT-V **9 Lectures**

Advanced Concepts in Blockchain: InterPlanetary File System (IPFS), Zero Knowledge Proofs, Self-Sovereign Identity, Blockchain with AI/ML, Blockchain Cloud Offerings. Case Studies: Banking and Financial Services, Energy and Utilities

Text Books:

1. Blockchain for Enterprise Application Developers‖, First Edition, 2020, Ambadas Tulajadas Choudhari, Arshad Sarfarz Ariff, Sham M R, Wiley, ISBN: 9788126599967.
2. Blockchain Technology, Chandramouli Subramanian, Asha A George, Abhilash K A, MeenaKarthikeya, 2020, Universities Press.ISBN: 978938921163, Year: 2020

Reference Books:

1. Blockchain Fundamental, 16th Edition, 2019, Dr. Ravindhar Vadapalli, BPB Publications, ISBN: 978-93-8728-449-4.
2. Blockchain Basics: A Non-Technical Introduction in 25 Steps, Daniel Drescher, (2023), Apress.

Web References (e-Resources):

1. https://archive.nptel.ac.in/courses/106/105/106105235/
2. https://hyperledger-fabric.readthedocs.io/en/release-1.4/build_network.html

CLOUD COMPUTING

[Professional Elective]

L	T	P	C
3	0	0	3

Course Code: 24IT11PE01

Course Outcomes At the end of the course students are able to:

CO1: **Summarize** the key dimensions and fundamental concepts of various computing paradigms. (L2)

CO2: **Outline** the different Cloud Computing models and Service Types. (L2)

CO3: **Classify** the Levels of Virtualization & Utilize the virtualization concept to chips, CPUs, Memory and data centres in Cloud Environments. (L2)

CO4: **Make use of** Cloud infrastructure of Google App Engine, AWS and MS-Azure to develop applications Using Cloud Platforms. (L3)

CO5: **Analyze** different Distributed Storage Systems and Databases to build HPC applications. (L4)

UNIT-I: **8 Lectures**

History of Computing Paradigms: Overview of Distributed Computing, Cluster Computing, Grid Computing, Ubiquitous Computing, Peer-to-Peer Computing. Distributed System Models and Enabling Technologies.

UNIT-II: **10 Lectures**

Introduction to Cloud Computing: Cloud Computing and Service Models- Public, Private, and Hybrid Clouds, Cloud Ecosystem and Enabling Technologies, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

UNIT-III: **10 Lectures**

Virtual Machines and Virtualization: Implementation Levels of Virtualization, VMM Design Requirements and Providers, Virtualization Support at the OS Level, Middleware Support for Virtualization, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management.

Case Study: Xen, a VMM Based on Paravirtualization.

UNIT-IV: **10 Lectures**

Public Cloud Platform-Architectures and Programming: Google App Engine (GAE). Amazon Web Services (AWS), Microsoft Windows Azure.

Service Oriented Architecture: REST, Publish Subscribe Model.

UNIT-V: **12 Lectures**

Storage Systems: Storage Models, File Systems, and Databases, Distributed File Systems, General Parallel File System, Google File Systems, Apache Hadoop.

Locks and Chubby: A Locking Service, Transaction Processing and NoSQL Databases, Big Table, Mega Store.

Case Studies:

The GrepTheWeb Application, Aneka Application of Maya Rendering Case Study

Textbook:

1. Kai Hwang, Geoffrey C. Fox, Jack K. Dongarra, Distributed and Cloud Computing: From parallel processing to Internet of Things, Morgen Kauffmann 2013.
2. Cloud Computing Theory and Practice, Dan C. Marinescu.

Reference Books:

1. Arshdeep Bagha & Vijay Madisetti Cloud Computing: A Hands-On Approach, University Press, 2022 Edition
2. Anthony T. Velte Toby J.Velte, Ph.D. Robert ElsenpeterCloud Computing: A Practical Approach Cloud Computing Bible, Barrie Sosinsk

Web References(e-Resources):

1. https://nptel.ac.in/courses/106/105/106105167/
2. https://www.coursera.org/specializations/cloud-computing
3. https://www.coursera.org/learn/introduction-to-cloud

DEEP LEARNING

[Professional Elective]

Course Code: 24CI11PE02

L	T	P	C
3	0	0	3

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

CO1: Summarize the basic principles of artificial neural networks. (L2)

CO2: Interpret the various difficulties in training deep neural networks. (L2)

CO3: Develop deep learning models to computer vision problems and time series analysis. (L3)

CO4: Build deep learning models for generative AI. (L3)

CO5: Apply deep learning techniques in reinforcement learning. (L3)

UNIT-I **9 Lectures**

Introduction to ANNs with Keras: Biological and artificial neurons, implementing multilayer perceptrons with Keras, fine-tuning neural network hyper parameters.

UNIT-II **9 Lectures**

Training Deep Neural Networks: The vanishing/exploding gradients problems, Glorot and He initializations, batch normalization, reusing pretrained models, faster optimizers, avoiding overfitting through regularization.

UNIT-III **14 Lectures**

CNNs and RNNs: The architecture of the visual cortex, convolutional layers, pooling layers, famous CNN architectures, object detection, fully convolutional networks, YOLO, semantic segmentation. Recurrent neurons and layers, training RNNs, forecasting a time series, stacking recurrent layers, using bidirectional RNNs, handling long sequences using LSTMs and GRUs.

UNIT-IV **9 Lectures**

Generative deep learning: Efficient data representations, stacked, convolutional, recurrent, denoising, sparse and variational autoencoders, generative adversarial networks, basic introduction to transformers and stable diffusion.

UNIT-V **9 Lectures**

Deep Reinforcement Learning: Essential theory of reinforcement learning, Markov decision processes, the optimal policy, essential theory of deep Q-learning networks, defining a DQN agent.

Text Books:

1. Aurélien Géron (2022), ―Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow,‖ 3rd Edition, O,,Reilly Publications.
2. Krohn et al (2020) , ―Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence,‖ Pearson Education.

Reference Books:

1. Francois Chollet (2021), ―Deep Learning with Python,‖ 2nd Edition.

Web References (e-Resources):

1. https://onlinecourses.nptel.ac.in/noc20_cs62/preview
2. https://www.coursera.org/specializations/deep-learning

DISTRIBUTED SYSTEMS

[Professional Elective]

Course Code: 24CI11PE03

L	T	P	C
3	0	0	3

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

CO1: **Outline** the important characteristics of Distributed Systems and salient features of Distributed Systems. (L2)

CO2: **Make use of** Inter process communication mechanisms with TCP and UDP protocols in Distributed systems. (L3)

CO3: **Apply** RMI and RPC for Remote Invocation in Distributed systems for Distributed Objects. (L3)

CO4: **Illustrate** the Operating systems facilities at the nodes of a Distributed Systems and Examine the different file management systems in distributed systems. (L2)

CO5: **Classify** various protocols for transaction and replication of distributed systems. (L2)

UNIT-I **10 Lectures**

Characterization of Distributed Systems: Introduction, Examples of Distributed Systems, Resource Sharing and the Web, Challenges.

System Models: Introduction, Architectural Models- Software Layers, System Architecture, Variations, Interface and Objects, Design Requirements for Distributed Architectures, Fundamental Models- Interaction Model, Failure Model, Security Model.

UNIT-II **10 Lectures**

Inter process Communication: Introduction, The API for the Internet Protocols- The Characteristics of Inter process communication, Sockets, UDP Datagram Communication, TCP Stream Communication; External Data Representation and Marshalling; Client Server Communication; **Group Communication**- IP Multicast- an implementation of group communication, Reliability and Ordering of Multicast.

UNIT-III **10 Lectures**

Distributed Objects and Remote Invocation: Introduction, Communication between Distributed Objects- Object Model, Distributed Object Modal, Design Issues for RMI, Implementation of RMI, Distributed Garbage Collection; Remote Procedure Call, Events and Notifications, Case Study: JAVA RMI.

UNIT-IV **12 Lectures**

Operating System Support: Introduction, The Operating System Layer, Protection, Processes and Threads– Address Space, Creation of a New Process, Threads.

Distributed File Systems: Introduction, File Service Architecture; Peer-to-Peer Systems: Introduction, Napster and its Legacy, Peer-to-Peer Middleware, Routing Overlays.

UNIT-V **8 Lectures**

Transactions & Replications: Introduction, System Model and Group Communication, Concurrency Control in Distributed Transactions, Distributed Dead Locks, Transaction Recovery; Replication- Introduction, Passive (Primary) Replication, Active Replication.

Text Books:

1. Distributed Systems- Concepts and Design‖, FourthEdition, George Coulouris, Jean Dollimore, Tim Kindberg, Pearson Publication.
2. Distributed Computing, Principles, Algorithms And systems, Ajay D Kshem kalyani, Mukesh Sighal, Cambridge.

Reference Books:

1. "Distributed Systems: Principles and Paradigms", 2nd Edition, 2006, Andrew S. Tanenbaum, Maarten Van Steen, Pearson Education, ISBN: 978-81-317-2523-2.
2. "Distributed and Cloud Computing: From Parallel Processing to the Internet of Things", 1st Edition, 2011, Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, Morgan Kaufmann,ISBN:978- 93-8093-918-0.

Web References (e-Resources):

1. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-824- distributed- systems-fall-2020/
2. https://www.distributed-systems.net/index.php/books/ds3/
3. https://www.coursera.org/browse/computer-science/cloud-computing
4. https://ieeexplore.ieee.org/Xplore/home.jsp

D A T A S C I E N C E

[Professional Elective]

Course Code: 24IA11PE01

L	T	P	C
3	0	0	3

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

CO1: **Summarize** the fundamental concepts of data science, including the types of data, the role of a data scientist, and real-world applications (L2)

CO2: **Utilize** machine learning techniques in Data Science with Python for effective modelling, validation, and real-world applications like malicious URL detection and recommender systems (L3)

CO3: **Apply** NoSQL and Hadoop for big data processing and real-world cases like loan risk assessment and disease diagnosis. (L3)

CO4: **Explain** the concepts and applications of graph databases using Neo4j and Cypher, and demonstrate text mining and analytics using Python libraries like NLTK and SQLite, with a case study on classifying Reddit posts (L2)

CO5: **Build** eco-friendly applications using object-oriented programming concepts, and build a prototype application of Data Science (L3)

UNIT-I: **10 Lectures**

Data Science Fundamentals: Structure, Objective, What is data?, Structured data, Unstructured data, Semi-structured data, What is data science?, What does a data scientist do?, Real-world use cases of data science.

Data Science process: Overview, defining goals and creating project charter, retrieving data, cleansing, integrating and transforming data, exploratory analysis, model building, presenting findings and building applications on top of them

UNIT-II: **10 Lectures**

Applications of machine learning in Data science, role of ML in DS, Python tools like sklearn, modelling process for feature engineering, model selection, validation and prediction, types of ML, semi-supervised learning. Handling large data: problems and general techniques for handling large data, programming tips for dealing large data, case studies on DS projects for predicting malicious URLs, for building recommender systems

UNIT-III: **10 Lectures**

NoSQL movement for handling Bigdata: Distributing data storage and processing with Hadoop framework, case study on risk assessment for loan sanctioning, ACID principle of relational databases, CAP theorem, base principle of NoSQL databases, types of NoSQL databases, case study on disease diagnosis and profiling.

UNIT-IV: **10 Lectures**

Tools and Applications of Data Science: Introducing Neo4j for dealing with graph databases, graph query language Cypher, Applications graph databases, Python libraries like nltk and SQLite for handling Text mining and analytics, case study on classifying Reddit posts.

UNIT-V: **10 Lectures**

Data Visualization and Prototype Application Development: Data Visualization options, crossfilter, the JavaScript Map Reduce library, creating an interactive dashboard with dc.js, Dashboard development tools, applying the DS process for respective engineering problem solving scenarios as a detailed case study

Textbook:

1. Davy Cielen, Arno D.B.Meysman, and Mohamed Ali, "Introducing to Data Science using Python tools", Manning Publications Co, Dreamtech press, 2016.
2. Prateek Gupta, "Data Science with Jupyter" BPB publishers, 2019 for basics

Reference Books:

1. Joel Grus, "Data Science from Scratch", OReilly, 2019.
2. Doing Data Science: Straight Talk from The Frontline, 1 st Edition, Cathy O'Neil and Rachel Schutt, O'Reilly, 2013

Web References(e-Resources):

1. Data science for engineers" https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs28/

ETHICAL HACKING

[Professional Elective]

Course Code: 24CT11PE02

L	T	P	C
3	0	0	3

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

CO1: **Explain** hacking concepts and penetration testing techniques to assess system vulnerabilities, thereby enhancing cyber security measures and system defences. (L2)

CO2: **Apply** foot printing and scanning techniques to analyse network systems and evaluate security vulnerabilities for effective penetration testing. (L3)

CO3: **Make use of** system hacking techniques, password-cracking methods, and understand malware operations to assess and mitigate security threats. (L3)

CO4: **Apply** sniffing, session hijacking techniques, analyse packet data, and **understand** social engineering methods to identify and mitigate security threats. (L3)

CO5: **Demonstrate** cryptography, steganography, and vulnerability assessment techniques to analyse system security and **apply** testing methods for identifying and mitigating risks. (L3)

UNIT-I — 8 Lectures

Introduction to Hacking: Hacking, Types and phases of hacking.

Introduction to Ports & Protocols: Ports, Protocols, Primary Network Types, Virtualization & Introduction to Kali Linux: Virtualization, Virtualization software, supported platforms, Introduction to Penetration Testing: Penetration test, Categories and Types of Penetration tests, Structure of Penetration Test Report.

UNIT-II — 8 Lectures

Foot printing: Foot printing, Types, Using ping and ns Lookup commands in Windows command line. **Scanning**: Scanning, Basics of Scanning, Basic Techniques of Scanning, Enumerating DNS using DNS, enum, performing flag scan using hping3.

UNIT-III — 10 Lectures

Hacking into System: System Hacking, Password Cracking, Default password databases, Manual and Automated Password Cracking, Process of System Hacking, Rootkits, Using Key Loggers.

Trojans & Backdoors: Trojans, Working of Trojan, Infection Techniques, Attack, Lifecycle and Classification of Virus, Worms, Virus Construction Kit.

UNIT-IV — 10 Lectures

Sniffing, Types of Sniffing-Active and passive sniffing Techniques, Packet Analysis, ARP Spoofing, session hijacking

Social Engineering: Social Engineering, Process, Identity Theft, Human and Computer Based Social Engineering Techniques, Phishing Process, Types of Phishing Attacks, Social Engineering Toolkit (SET).

UNIT-V **10 Lectures**

Vulnerability Assessment: Vulnerability, The Open Web Application Security Project (OWASP), Prevention, Damn Vulnerable Web Application (DVWA), installation and testing of DVWA

Advanced Hacking: Web Application & Web Server Penetration Testing, Denial of Service, Wireless Network Exploitation

Case study: The students are advised to perform End-to-End Penetration Testing on DVWA (Damn Vulnerable Web Application).

Text Books:

1. Hacking: Be a Hacker with Ethics, Harsh Bothra, Khanna Publications, 2019.
2. Ethical Hacking and Penetration Testing Guide, Rafay Baloch, 2014.

Reference Books:

1. Kali Linux Wireless Penetration Testing Beginner's Guide, Vivek Ramachandran, Cameron Buchanan, Packt Publishing, 2015.
2. SQL Injection Attacks and Defense, 1st Edition, Justin Clarke-Salt, Syngress Publication.
3. Mastering Modern Web Penetration Testing, Prakhar Prasad, Packt Publishing, October 2016.

Web References:

1. https://onlinecourses.nptel.ac.in/noc22_cs13/preview

GENERATIVE AI MODELS

[Professional Elective]

Course Code: 24CT11PE03

L	T	P	C
3	0	0	3

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

CO1: **Summarize** large language models' architecture and pre-training techniques. (L2)

CO2: **Make Use of** the GPT model for natural language processing tasks. (L3)

CO3: **Apply** the BERT model for natural language processing tasks. (L3)

CO4: **Utilize** LSTM, Transformers, GANs, and VAEs for text and image generation, including fine-tuning models for real-world applications. (L3)

CO5: **Apply** LSTM and Transformers for music generation and composition. (L3)

UNIT-I **10 Lectures**

Introduction to Large Language Models: Overview of Generative AI and Large Language Models. Basics of attention mechanisms and Transformer architecture. Pre- training techniques and transfer learning strategies.

UNIT-II **10 Lectures**

GPT Models and Applications: Study of GPT architecture and variants. Applications of GPT models in text generation and dialogue systems. Case study-based implementation of GPT-based tasks. GPT-based chatbot enhances E-Shop's customer support service

UNIT-III **8 Lectures**

BERT and Advanced Techniques: Understanding BERT architecture and pre-training objectives. Fine-tuning BERT for downstream NLP tasks. Exploration of advanced Transformer architectures and techniques

UNIT-IV **12 Lectures**

Text Generation with Generative AI: Introduction to Text Generation, LSTM-based Text Generation, Transformer-based Text Generation, Fine-Tuning Language Models, and Text Generation Applications

Image Generation with Generative AI: Introduction to Image Generation, Implementing GANs for Image Generation Training and Fine-Tuning GANs, Generating Images with VAEs, Advanced Techniques in Image Generation, and Image and Video Generation Applications.

UNIT-V **10 Lectures**

Music Generation with Generative AI:

Introduction to Music Generation, Music Representation, and LSTM-based Music Generation. Transformer-based Music Generation, Evaluation and Fine-Tuning, Music Composition Applications.

Text Books:

1. Generative AI for Everyone: Understanding the Essentials and Applications of This Break through Technology". Altaf Rehmani.
2. Generative AI with Python and TensorFlow 2: Create images, text, and music with VAEs, GANs, LSTMs, Transformer models, Joseph Babcock and Raghav Bali, 2024.

Reference Books:

1. Generative Adversarial Networks Cookbook: Over 100 recipes to build generative models using Python, TensorFlow, and Keras by Josh Kalin.
2. Generative AI in Software Development: Beyond the Limitations of Traditional Coding Jesse Sprinter, 2024.
3. Deep Learning- Ian Goodfellow, Yoshua Bengio and Aaron Courvile, MIT Press, 2016

Web References (e-Resources):

1 https://elearn.nptel.ac.in/shop/iit-workshops/completed/leveraging-generative-ai-for- teaching programming-course s/?v=c86ee0d9d7ed

PRINCIPLES OF COMPLIER DESIGN

[Professional Elective]

Course Code: 24IA11PE02

L	T	P	C
3	0	0	3

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

CO1: **Summarize** the phases of the Compiler and role of Lexical Analyzer. (L2)

CO2: **Make use of** different parsing techniques for the construction of parse tree. (L3)

CO3: **Explain** the semantic analysis Schemes and Intermediate code generation. (L2)

CO4: **Outline** the various Machine-Independent and dependent Optimization techniques of compiler. (L2)

CO5: **Utilize** the code generation techniques and role of the symbol table in the compiler design. (L3)

UNIT – I **8 Lectures**

Introduction to Compiler: Language Processors, the structure of a compiler, the science of building a compiler, programming language basics**,** Compiler tools, Boot strapping**.**

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator LEX, Design of a Lexical-Analyzer Generator.

UNIT-II **12 Lectures**

Syntax Analyzer: Introduction to Context-Free Grammars, Introduction to Parsers, Top-Down Parsing techniques: Brute force parsing, Recursive Descent Parsing, Predictive Parsing. Bottom-Up Parsing: Shift reduce parsing, Simple LR, More Powerful LR (K) Parsers.

UNIT-III **10 Lectures**

Semantic Analysis & Intermediate code Generation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax-Directed Translation, Syntax-Directed Translation Schemes, and Intermediate Code Generation, Three Address Code-Translation of Expressions, Control Flow.

UNIT-IV **10 Lectures**

Code Optimization: The Principal Sources of Optimization, Introduction to Data-Flow Analysis, Foundations of Data-Flow Analysis. Loop Optimization, Basic Blocks & Flow Graphs, Local Optimization, DAG Representation of Basic Blocks, Applications of DAG, Unreachable Code Elimination, Dead Code Elimination, Machine Dependent Optimization: Peep-Hole Optimization.

UNIT-V **10 Lectures**

Code Generation: Issues in Code Generation, Input to Code Generator, Instruction Selection, Register Allocation, Simple Target Machine Model, Program and Instruction Costs, Register allocation & Assignments, Code Generation Algorithm.

Symbol Table & Run time environment: Structure of Symbol Table, declarations, Run time Environment, Stack allocation of space, Heap Management, Garbage collection.

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S Lam, Ravi Sethi, Jeffry D. Ullman, Pearson.
2. Compiler Construction-Principles and Practice, Kenneth C Louden, Cengage Learning. ISBN: 9788131501320

Reference Books:

1. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
2. Louden: —Compiler Construction, Principles & Practice‖, 1st Edition, Thomson Press, 20063.
3. The Theory and Practice of Compiler writing, J. P. Tremblay and P. G. Sorenson, TMH
   Writing compilers and interpreters, R. Mak, 3rd edition, Wiley student edition.

Web References (e-Resources):

1. https://nptel.ac.in/courses/106104123
2. https://cse.iitkgp.ac.in/~rkumar/advcc/resources.html

IMAGE PROCESSING

[Professional Elective]

Course Code: 24IT11PE02

L	T	P	C
3	0	0	3

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

CO 1: Illustrate the fundamental steps in Digital Image processing and basic operations on image

CO 2: Explain the spatial and frequency domain preprocessing techniques

CO 3: Summarize the phases of morphological operations for image segmentation

CO 4: Outline the various image compression techniques

CO 5: Explain the methods for object detection and classification

UNIT-I: Introduction to Image processing — 8 Lectures

Digital image fundamentals, fundamental steps in digital image processing, Digital image representation, uses of digital image processing, imaging sensing and acquisition, image sampling and quantization. Some basic relationships between pixels, grey scale, RGB, HSI images, Pseudo colour image processing, Introduction to mathematical tools used in Digital image processing.

UNIT-II: Preprocessing — 12 Lectures

Noise in the images, Noise models, noise reduction using Spatial Filtering and frequency domain filtering. Basic intensity transformation functions histogram and histogram equalization, fundamentals of spatial filtering, smoothing spatial filters, sharpening spatial filters. Transformation using Discrete Fourier transforms. Filtering in the frequency domain, image smoothing using frequency domain filters, Selective filtering, Implementation.

UNIT-III: Image segmentation — 10 Lectures

Fundamentals, point, line, edge detection thresholding, region –based segmentation, segmentation using Morphological watersheds, the use of motion in segmentation. Morphological image processing: Erosion and dilation, opening and closing, the Hit-or-miss transformation, some Basic Morphological algorithms, grey–scale morphology.

UNIT-IV: Image compression — 10 Lectures

Need for Image compression, Redundancy in images, coding redundancy, spatial and temporal redundancy, measuring image information, compression standards. Compression methods: Run length coding, Huffman coding, Arithmetic coding, and Transform based compression.

UNIT-V: Object recognition **8 Lectures**

Need for an Object-recognition System, Automated Object-recognition Systems, selection of Measurement Parameters. Approaches to Object Recognition: Bayes' Parametric Classification, Template based detection, Neural-network Approach to Object Recognition, Multilayer perceptron and applications of object recognition.

Textbooks:

1. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edition, Prentice Hall, 2008.
2. Jayaraman, S. Esakkirajan, and T. Veerakumar, *Digital Image Processing*, Tata McGraw Hill Education.

Reference Books:

1. Anil K.Jain, "*Fundamentals of Digital Image Processing*", Prentice Hall of India, 9th Edition, Indian Reprint, 2002.
2. John C. Russ, J. Christian Russ,*Introduction to Image Processing & Analysis*- CRC Press, 2010

Web References:

1. https://archive.nptel.ac.in/courses/117/105/117105135/
2. https://sites.ecse.rpi.edu/~rjradke/improccourse.html

INFORMATION RETRIEVAL AND SYSTEMS

[Professional Elective]

Course Code: 24IT11PE03

L	T	P	C
3	0	0	3

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

CO1: Summarize information retrieval systems and the various data structures and algorithms used in it

CO2: Outline the different techniques of storing and retrieving documents using Inverted Files and Signature Files.

CO3: Construct a DFA for Lexical analysis to identify index terms and apply stemming algorithms on it.

CO4: Build textual information into Thesaurus and apply various string search algorithms.

CO5: Examine the usage of web search engine for crawling the data.

UNIT-I: **8 Lectures**

Introduction to Information storage and retrieval systems: Domain Analysis f IR systems, IR and other types of Information Systems, IR System Evaluation.

Introduction to Data structures and algorithms related to Information Retrieval: Basic Concepts, Data structures, Algorithms.

UNIT-II: **10 Lectures**

Inverted Files: Introduction, Structures used in Inverted Files, Building an Inverted files using a sorted array, Modifications to the Basic Techniques.

Signature Files: Introduction, Concepts of Signature files, Compression, Vertical Partitioning, Horizontal Partitioning.

UNIT-III: **12 Lectures**

Lexical Analysis and Stop lists: Introduction, Lexical Analysis, Stop lists.

Stemming Algorithms: Introduction, Types of Stemming algorithms, Experimental Evaluations of Stemming, Stemming to Compress Inverted Files.

UNIT-IV: **10 Lectures**

Thesaurus Construction: Introduction, Features of Thesauri, Thesaurus Construction, Thesaurus construction from Texts.

String Searching Algorithms: Introduction, Preliminaries, The Naive Algorithm, The Knutt Morris-Pratt Algorithm, The Boyer-Moore Algorithm.

UNIT-V: **10 Lectures**

Web Search Engine: Web search basics, Web characteristics, Index size and estimation, Web crawling and indexes, Distributing indexes.

Textbook:

1. Raghavan, and H. Schutze Information Retrieval Data Structures and Algorithms ,
2. C. D. Manning, P, Introduction to Information Retrieval, Cambridge University Press (2008).

Reference Books:

1. Ricardo Baeza-Yates, Neto Modern Information Retrieval, PEA,2007.
2. Gerald J.Kowalski, Mark T.Maybury, "Information storage and Retrieval systems: theory and implementation", 2nd Edition, kluwer academic publishers, 2009.

Web References:

1.http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

HADOOP AND BIG DATA ANALYTICS

[Professional Elective]

Course Code: 24CI11PE04

L	T	P	C
3	0	0	3

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

CO1: **Outline** the building blocks of Hadoop. (L2)

CO2: **Experiment with** writing basic Map Reduce programs. (L3)

CO3: **Demonstrate** PIG Architecture and Develop PIG scripts. (L2)

CO4: **Apply** Hive queries to create, update, delete, and manage data in databases. (L3)

CO5: **Analyse** Spark architecture, apply transformations, manage RDDs, and perform operations for efficient data processing. (L4)

UNIT-I **10 Lectures**

Introduction to Big Data and Hadoop: Introduction to Big Data Platform, Challenges of Conventional Systems, History of Hadoop, building blocks of Hadoop (Namenode, Datanode,Secondary Namenode, JobTracker, TaskTracker), Hadoop Distributed File System- The Design oHDFS, HDFS Concepts, Basic Filesystem Operations, Anatomy of a Map Reduce Job run, Map Reduce Types and Formats.

UNIT-II **10 Lectures**

Writing MapReduce Programs: A Weather Dataset, Basic programs of Hadoop MapReduc-Driver code, Mapper code, Reducer code, RecordReader, Combiner, Partitioner

UNIT-III **8 Lectures**

Pig: Hadoop Programming Made Easier: Admiring the Pig Architecture, Going with the Pig Latin Application Flow, Working through the ABCs of Pig Latin, Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin.

UNIT-IV **10 Lectures**

Hive: Introduction to Hive, data types, file formats, HiveQL data definition, HiveQL data manipulation, joins, Window functions, Table partitioning, Bucketing, Indexing, User-Defined Functions.

UNIT-V **12 Lectures**

Apache spark- Introduction to Spark - Spark,,s Basic Architecture, Transformations- Narrow and Wide, lazy evaluation, Actions, Advantages over Hadoop

Basic Structured Operations: Schemas, Records and Rows, DataFrame Transformations,

Resilient Distributed Datasets (RDDs) - Creating RDDs, Manipulating RDDs, Transformations, Actions, Saving Files, Classification Models in MLlib: Naïve Bayes, Random Forest.

Text Books:

1. Hadoop: The Definitive Guide by Tom White, 3rd Edition, O,,reilly
2. Hadoop for Dummies by Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk,Bruce Brown, Rafael Coss
3. SPARK: The Definitive Guide, Bill Chambers & Matei Zaharia, O'Reilley, 2018 Edition

Reference Books:

1. Programming Hive, O'Reilley, E. Capriolo, D. Wampler, and J. Rutherglen, 2012

2. Programming Pig, O'Reilley, Alan Gates, 2011

Web References (e-Resources):

1. https://archive.nptel.ac.in/courses/106/104/106104189/
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0126052684230082561692_shared/overview
3. https://sparkbyexamples.com/

MOBILE APPLICATION DEVELOPMENT

[Professional Elective]

Course Code: 24CT11PE04

L	T	P	C
3	0	0	3

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

CO1: **Explain** the Android development framework. (L2)

CO2: **Demonstrate** usage of user interface components and layouts. (L3)

CO3: **Utilize** Image Views to display pictures dynamically in Android applications. (L3)

CO4: **Utilize** networking functionalities to fetch and send data over the intent. (L3)

CO5: **Build** APK files and configure them for different Android devices. (L3)

UNIT-I **12 Lectures**

Getting Started with Android Programming: Introduction to Android, Obtaining the Required Tools Activities and Intents**:** Software Economics, pragmatic software cost estimation**.**

Improving Software Economics: Understanding Activities, Linking Activities Using Intents, Calling Built-In Applications Using Intents,

UNIT-II **12 Lectures**

Getting to Know the Android User Interface: Understanding the Components of a Screen, Adapting to Display Orientation, Managing Changes to Screen Orientation, Creating the User Interface Programmatically.

Designing Your User Interface Using Views: Basic Views, Picker Views, List Views.

UNIT-III **8 Lectures**

Fragments: Creating Fragments with Java Code . Creating Special Fragments.

Displaying Pictures and Menus with Views: Using Image Views to Display Pictures, Using Menus with Views.

UNIT-IV **8 Lectures**

Messaging and Networking: SMS Messaging, Sending E-Mail, Networking.

Data Persistence: Saving and Loading User Preferences, Persisting Data to Files, Creating and Using Databases

UNIT-V **10 Lectures**

Publishing Android Applications: Preparing for Publishing, Deploying APK Files, Publishing on the Android Market.

Text Books:

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013.
2. Android™ Programming Unleashed, B.M. Harwani, 1st Edition, Pearson Education, 2013.

Reference Books:

1. Reto Meier , Professional Android 4 Application Development, Wiley India, (Wrox) , 2012.
2. James C. Sheusi, Android Application Development for Java Programmers, Cengage Learning, 2013.

Web References (e-Resources):

1. https://developer.android.com/docs
2. https://www.technosip.com/mobile-application-development/
3. https://www.denimgroup.com/media/pdfs/MobileDevReference.pdf

NATURAL LANGUAGE PROCESSING (NLP)

[Professional Elective]

Course Code: 24AI11PE03

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

CO1: Illustrate key NLP methodologies, including rule-based, statistical, and deep learning-based approaches. (L2)

CO2: Utilize neural embeddings and deep learning techniques to enhance text classification performance. (L3)

CO3: Make use of Transformer-based models in combination with traditional NLP techniques to optimize Information Extraction (IE) performance. (L3)

CO4: Build chatbot applications using various dialog system components and frameworks like Rasa NLU. (L3)

CO5: Formulate the effectiveness of NLP applications in different domains. (L4)

UNIT-I: **10 Lectures**

NLP: A Primer:

NLP in the Real World, Building Blocks of Language, Machine Learning, Deep Learning, and NLP: An Overview, Approaches to NLP, An NLP Walkthrough: Conversational Agents.

NLP Pipeline:

Data Acquisition, Text Extraction and Cleanup, Pre-Processing, Feature Engineering, Modelling, Evaluation, Post-Modelling Phases.

Case Study: Customer Obsession Ticketing Assistant (COTA)

UNIT-II: **10 Lectures**

Text Representation:

Vector Space Models, Basic Vectorization Approaches, Distributed Representations, Distributed Representations Beyond Words and Characters, Universal Text Representations, Visualizing Embeddings.

Text Classification:

Applications, A Pipeline for Building Text Classification Systems, One Pipeline, Many Classifiers, Using Neural Embeddings in Text Classification, Deep Learning for Text Classification, Interpreting Text Classification Models, Learning with No or Less Data and Adapting to New Domains.

Case Study: A corporate ticketing system.

UNIT-III: **10 Lectures**

Information Extraction:

IE Applications, IE Tasks, The General Pipeline for IE, Keyphrase Extraction, Named Entity Recognition, Named Entity Disambiguation and Linking, Relationship Extraction, Other Advanced IE Tasks.

Transformer Anatomy:

The Transformer Architecture, The Encoder, The Decoder, Meet the Transformers.

UNIT-IV: **10 Lectures**

Chatbots:

Applications, A Taxonomy of Chatbots, A Pipeline for Building Dialog Systems, Dialog Systems in Detail, Deep Dive into Components of a Dialog System, Other Dialog Pipelines, Rasa NLU.

UNIT-V: **14 Lectures**

NLP Applications:

Search and Information Retrieval, Topic Modelling, Text Summarization, Recommender Systems for Textual Data, Machine Translation, Question-Answering Systems.

Textbook:

1. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta and Harshit Surana, “Practical Natural Language Processing”, O’Reilly Media Inc., 2021, ISBN: 978-93- 8588-918-9.
2. Lewis Tunstall, Leandro von Werra, & Thomas Wolf, Natural Language Processing withTransformers ISBN: 978-1098136789.

Reference Books:

1. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, 2ndEdition, Daniel Jurafsky, James H. Martin -Pearson Publication,2014.

Web References:

1. https://onlinecourses.nptel.ac.in/noc23_cs45/preview
2. https://www.coursera.org/specializations/natural-language-processing

SOFT COMPUTING

[Professional Elective]

Course Code: 24CT11PE05

L	T	P	C
3	0	0	3

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

CO1: **Summarize** various soft computing techniques in order to solve problems effectively and efficiently. (L2)

CO2: **Outline** appropriate learning rules for each of the architectures based supervised neural networks with its applications. (L2)

CO3: **Outline** appropriate learning rules for each of the architectures based unsupervised neural networks and third-generation neural networks with its applications. (L2)

CO4: **Demonstrate** Fuzzy logic concepts to recognize, represent, manipulate, interpret, and use data and information that is vague and lacks certainty. (L2)

CO5: **Solve** optimization problems using genetic algorithms and **Develop** hybrid system using the principles of soft computing techniques in various applications. (L3)

UNIT-I **8 Lectures**

Introduction: Neural Networks, Application Scope of Neural Networks, Fuzzy Logic, Genetic Algorithm, Hybrid Systems, Soft Computing.

UNIT-II **12 Lectures**

Artificial Neural Network: Fundamental Concept, Evolution of Neural Networks, Basic Models of Artificial Neural Network, Important Terminologies of ANNs, McCulloch–Pitts Neuron, Linear Separability, Hebb Network.

Supervised Learning Network: Perceptron Networks, Adaptive Linear Neuron (Adaline), Back- Propagation Network, Autoassociative Memory Network, Heteroassociative Memory Network

UNIT-III **10 Lectures**

Unsupervised Learning Networks: Fixed Weight Competitive Nets, Kohonen Self- Organizing Feature Maps, Learning Vector Quantization, Counter propagation Networks, Adaptive Resonance Theory Network.

Third-Generation Neural Networks: Spiking Neural Networks, Convolutional Neural Networks

UNIT-IV **10 Lectures**

Fuzzy Set Theory: Fuzzy Versus Crisp, Crisp Sets, Fuzzy Sets, Crisp Relations, Fuzzy Relations **Fuzzy Systems:** Crisp Logic, Predicate Logic, Fuzzy Logic, Fuzzy Rule based System, Defuzzification Methods

UNIT-V **10 Lectures**

Fundamentals of Genetic Algorithms: Genetic Algorithms, Basic concepts, Creation of Offsprings, Working Principle, Encoding, Fitness function, Reproduction.

Hybrid Soft Computing Techniques: Neuro-Fuzzy Hybrid Systems, Genetic Neuro-Hybrid Systems, Genetic Fuzzy Hybrid and Fuzzy Genetic Hybrid Systems.

Text Books:

1. Principles of Soft Computing‖, 3rd Edition, S. N. Sivanandam, S. N. Deepa,Wiley India Private Ltd., 2019, ISBN: 978-81-265-7713-2.
2. Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications‖, S. Rajsekaran and G. A. Vijayalakshmi Pai,Prentice Hall of India.

Reference Books:

1. Neuro-Fuzzy and Soft Computing‖, J. S. R. Jang, C.T. Sun and E. Mizutani, PHI / Pearson Education 2004.
2. Neural Networks Comprehensive Foundation‖ Second Edition, Simon Haykin, Pearson Education, 2005.

Web References (e-Resources):

1. https://archive.nptel.ac.in/courses/106/105/106105173/

UNIX AND SHELL PROGRAMMING

[Professional Elective]

Course Code: 24IT11PE04

L	T	P	C
3	0	0	3

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

CO1: Outline Unix architecture, commands, and file handling operations

CO2: Analyse redirection, pipes, and filters to manipulate data in Unix.

CO3: Explain text-processing scripts using grep, sed, and awk.

CO4: Identify Korn shell scripts for automation.

CO5: Apply C shell scripts for process automation and environment management.

UNIT-I: **8 Lectures**

Introduction to Unix: The Architecture of Unix, Features of Unix, locating commands, Internal and external commands, General-Purpose Utilities, The File System and handling ordinary files.

UNIT-II: **12 Lectures**

Introduction to Shells: Unix Session, Standard Streams, Redirection, Pipes, Tee Command, Command Execution, Command- Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment Customization.

Filters: Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count characters, Words or Lines, Comparing Files.

UNIT-III: **10 Lectures**

Text processing in Unix/Linux: Grep: Operation, grep Family, Searching for File Content.

Sed: Scripts, Operation, Addresses, commands, Applications, grep and sed.

awk: Execution, Fields and Records, Scripts, Operations, Patterns, Actions, Associative Arrays, String. Functions, String Functions, Mathematical Functions, User – Defined Functions, Using System commands, in awk, Applications, awk and grep, sed and awk.

UNIT-IV: **10 Lectures**

Interactive Korn Shell: Korn Shell Features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, Options, Startup Scripts, Command History, Command Execution Process.

Korn Shell Programming: Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples

UNIT-V: **10 Lectures**

Interactive C Shell: C shell features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, On-Off Variables, Startup and Shutdown Scripts, Command History, Command Execution Scripts.

C Shell Programming: Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.

Textbook:

1. Behrouz A. Forouzan, Richard F. Gilberg. Thomson Unix and shell Programming.
2. Sumitabha Das, Mc Graw Hill Unix Concepts and Applications, 4th Edition.

Reference Books:

1. Graham Glass, King Ables, Pearson Education Unix for programmers and users, 3rd edition,
2. Kernighan and Pike, PHI. / Pearson Education Unix programming environment, S Rosen, Host, Klee, Farber, Rosinski. The Complete Reference Unix, Second Edition, TMH.

Web References:

1. https://archive.nptel.ac.in/courses/117/106/117106113/#

INTRODUCTION TO BLOCKCHAIN TECHNOLOGY

[Open Elective]

Course Code: 24IT11EL01

L	T	P	C
3	0	0	3

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

CO1: **Summarize** the Blockchain ecosystem, properties, advantages and limitations. (L2)

CO2: **Explain** Blockchain‘s decentralized and distributed nature. (L2)

CO3: **Interpret** the components and transaction flow of Hyperledger Fabric within private blockchain ecosystems. (L2)

CO4: **Apply** Intelligent techniques in Blockchain. (L3)

CO5: **Utilize** advanced blockchain concepts to design and implement various blockchain applications. (L3)

UNIT-I: **10 Lectures**

Blockchain and its evolution: Introduction to Blockchain, Scenarios, Characteristics of Blockchain, History of Blockchain, Stages in Blockchain Evolution, Restrictions on Sharing Ledgers, Block structure, Chaining of Blocks, Security on Blockchain, Type of players in Blockchain ecosystem.

UNIT-II: **12 Lectures**

Distributed Consensus: Decentralized Blockchain, Merkle Tree, Consensus: Byzantine General Problem, Real Need for Mining, Proof of Work (PoW), Proof of Stake (PoS), Proof of Burn (PoW), Proof of Concept (PoC), Practical Byzantine Fault Tolerance (EPBFT).

UNIT-III: **10 Lectures**

Ethereum and Smart Contracts: Ethereum ecosystem, Ethereum development, Ethereum Tool Stack, Ethereum virtual machine, How Mining Works, Ethereum Workflow, Smart Contract programming, Integrated Development Environment, Ganache, Ethereum Accounts, My Ether Wallet, Ether Scan, Decentralized application, Ethereum Networks/ Environments, Metamask, Tuna Fish Use case Implementation.

UNIT-IV: **9 Lectures**

Private Blockchain with Hyperledger: Introduction, Private Blockchain Ecosystem, Hyperledger Fabric Transaction Flow, FabCar Use case Implementation, Invoking Chaincode functions using Client Application.

UNIT-V: **9 Lectures**

Advanced Concepts in Blockchain: InterPlanetary File System (IPFS), Zero Knowledge Proofs, Self-Sovereign Identity, Blockchain with AI/ML, Blockchain Cloud Offerings. Case Studies: Banking and Financial Services, Energy and Utilities.

Textbook:

1. Ambadas Tulajadas Choudhari, Arshad Sarfarz Ariff, Sham M R, "Blockchain for Enterprise Application Developers", First Edition, 2020, Wiley, ISBN: 9788126599967.
2. Chandramouli Subramanian, Asha A George, Abhilash K A and MeenaKarthikeyan. "Blockchain Technology", 2020, Universities Press. ISBN: 9789389211634 | Year: 2020

Reference Books:

1. Dr. Ravindhar Vadapalli, "BLOCKCHAIN FUNDAMENTALS TEXTBOOK", 16th Edition, 2019, BPB Publications, ISBN: 978-93-8728-449-4.
2. CFE, C. (2023). Blockchain Basics: A Non-Technical Introduction in 25 Steps. The CPA Journal, 93(3/4), 13-13.

Web References:

1. https://archive.nptel.ac.in/courses/106/105/106105235/
2. https://hyperledger-fabric.readthedocs.io/en/release-1.4/build_network.html

INTRODUCTION TO CLOUD COMPUTING

[Open Elective]

Course Code: 24IT11EL02

L	T	P	C
3	0	0	3

Course Outcomes At the end of the course students are able to:

CO1: **Summarize** the key dimensions and fundamental concepts of various computing paradigms. (L2)

CO2: **Outline** the different Cloud Computing models and Service Types. (L2)

CO3: **Classify** the Levels of Virtualization & Utilize the virtualization concept to chips, CPUs, Memory and data centres in Cloud Environments. (L2)

CO4: **Make use of** Cloud infrastructure of Google App Engine, AWS and MS-Azure to develop applications Using Cloud Platforms. (L3)

CO5: **Analyze** different Distributed Storage Systems and Databases to build HPC applications. (L4)

UNIT-I: **8 Lectures**

History of Computing Paradigms: Overview of Distributed Computing, Cluster Computing, Grid Computing, Ubiquitous Computing, Peer-to-Peer Computing. Distributed System Models and Enabling Technologies.

UNIT-II: **10 Lectures**

Introduction to Cloud Computing: Cloud Computing and Service Models- Public, Private, and Hybrid Clouds, Cloud Ecosystem and Enabling Technologies, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

UNIT-III: **10 Lectures**

Virtual Machines and Virtualization: Implementation Levels of Virtualization, VMM Design Requirements and Providers, Virtualization Support at the OS Level, Middleware Support for Virtualization, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management.

Case Study: Xen, a VMM Based on Para virtualization.

UNIT-IV: **10 Lectures**

Public Cloud Platform-Architectures and Programming: Google App Engine (GAE). Amazon Web Services (AWS), Microsoft Windows Azure.

Service Oriented Architecture: REST, Publish Subscribe Model.

UNIT-V: **12 Lectures**

Storage Systems: Storage Models, File Systems, and Databases, Distributed File Systems, General Parallel File System, Google File Systems, Apache Hadoop.

Locks and Chubby: A Locking Service, Transaction Processing and NoSQL Databases, Big Table, Mega Store.

Case Studies:

The Grep The Web Application, Aneka Application of Maya Rendering Case Study

Textbook:

1. Kai Hwang, Geoffrey C. Fox, Jack K. Dongarra, Distributed and Cloud Computing: From parallel processing to Internet of Things, Morgen Kauffmann 2013.
2. Cloud Computing Theory and Practice, Dan C. Marinescu.

Reference Books:

1. Arshdeep Bagha & Vijay Madisetti Cloud Computing: A Hands-On Approach, University Press, 2022 Edition.
2. Anthony T. Velte Toby J.Velte, Ph.D. Robert ElsenpeterCloud Computing: A Practical Approach Cloud Computing Bible, Barrie Sosinsk

Web References:

1. https://nptel.ac.in/courses/106/105/106105167/
2. https://www.coursera.org/specializations/cloud-computing
3. https://www.coursera.org/learn/introduction-to-cloud

INFORMATION RETRIEVAL SEARCH ENGINES

[Open Elective]

Course Code: 24IT11EL03

L	T	P	C
3	0	0	3

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

CO1: Summarize information retrieval systems and the various data structures and algorithms used in it

CO2: Outline the different techniques of storing and retrieving documents using Inverted Files and Signature Files.

CO3: Construct a DFA for Lexical analysis to identify index terms and apply stemming algorithms on it.

CO4: Build textual information into Thesaurus and apply various string search algorithms.

CO5: Examine the usage of web search engine for crawling the data.

UNIT-I: **8 Lectures**

Introduction to Information storage and retrieval systems: Domain Analysis f IR systems, IR and other types of Information Systems, IR System Evaluation.

Introduction to Data structures and algorithms related to Information Retrieval: Basic Concepts, Data structures, Algorithms.

UNIT-II: **10 Lectures**

Inverted Files: Introduction, Structures used in Inverted Files, Building an Inverted files using a sorted array, Modifications to the Basic Techniques.

Signature Files: Introduction, Concepts of Signature files, Compression, Vertical Partitioning, Horizontal Partitioning.

UNIT-III: **12 Lectures**

Lexical Analysis and Stop lists: Introduction, Lexical Analysis, Stop lists.

Stemming Algorithms: Introduction, Types of Stemming algorithms, Experimental Evaluations of Stemming, Stemming to Compress Inverted Files.

UNIT-IV: **10 Lectures**

Thesaurus Construction: Introduction, Features of Thesauri, Thesaurus Construction, Thesaurus construction from Texts.

String Searching Algorithms: Introduction, Preliminaries, The Naive Algorithm, The Knutt Morris-Pratt Algorithm, The Boyer-Moore Algorithm.

UNIT-V: **10 Lectures**

Web Search Engine: Web search basics, Web characteristics, Index size and estimation, Web crawling and indexes, Distributing indexes.

Textbook:

1. Raghavan, and H. Schutze Information Retrieval Data Structures and Algorithms ,
2. C. D. Manning, P, Introduction to Information Retrieval, Cambridge University Press (2008).

Reference Books:

1. Ricardo Baeza-Yates, Neto Modern Information Retrieval, , PEA,2007.
2. Gerald J.Kowalski, Mark T.Maybury, "Information storage and Retrieval systems: theory and implementation", 2nd Edition, kluwer academic publishers, 2009.

Web References:

1.http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf