MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(Deemed to be University)
MADANAPALLE
www.mits.ac.in

Estd 1998

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)
Course Structure
&
Detailed Syllabi

For the students admitted to

B. Tech. Regular Four Year Degree Programme during the Academic Year 2025-26

and

B. Tech. Lateral Entry Scheme during the Academic Year 2026-27



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(Deemed to be University)
MADANAPALLE
B. Tech Four Year Curriculum Structure
Branch: COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Total Credits	160 Credits for 2025 (Regular) & 120 Credits 2026 (Lateral Entry) Admitted Batch onwards

I. Induction Program and Holistic Development Activities

Sl.No	Title	Duration
1	Induction Program (Mandatory)	Three weeks' duration at the start of First Year

R25 - Curriculum Structure

I Year I Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	BSC	25BAMATTC02	Calculus for Computer Science	3	0	0	3	3
2	BSC	25BACHETC03	SMART Chemistry	3	0	0	3	3
3	ESC	25BACIVTC02	Introduction to Environment and Sustainability	2	0	0	2	2
4	ESC	25BAEEETC03	Digital Design	3	0	0	3	3
5	ESC	25BACSETC01	C Programming for Problem Solving	3	0	0	3	3
6	ESC	25BAMECEC01	Engineering Graphics	2	0	2	4	3
7	BSC	25BACHELC03	Chemistry Laboratory	0	0	2	2	1
8	ESC	25BACSELC01	C Programming for Problem Solving Laboratory	0	0	2	2	1
9	ESC	25BACOMLC01	Engineering Skills Laboratory	0	0	2	2	1
10	EEC		**NSDC - Certification Course – I *** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**16**	**0**	**10**	**26**	**20**

I Year II Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	AEC	25BAENGTC01	Professional Communication	2	0	0	2	2
2	HSMC	25BAHUMTC01	Human Values and Professional Ethics	2	0	0	2	2
3	BSC	25BAMATTC05	Linear Algebra	3	0	0	3	3
4	BSC	25BAPHYTC03	Modern Physics for Quantum Technologies	3	0	0	3	3
5	ESC	25BACSETC03	Data Structures and Algorithms	3	0	0	3	3
6	PCC	25BACAMTC01	Fundamentals of Intelligent Systems	3	0	0	3	3
7	AEC	25BAENGLC01	Professional Communication Laboratory	0	0	2	2	1
8	BSC	25BAPHYLC01	Physics Laboratory	0	0	2	2	1
9	ESC	25BACSELC02	IT Essentials Laboratory	0	0	2	2	1
10	ESC	25BACSELC03	Data Structures and Algorithms Laboratory	0	0	2	2	1
11	EEC		**NSDC - Certification Course – II *** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**16**	**0**	**10**	**26**	**20**

(L = Lecture, T = Tutorial, P = Practical, C = Credit)

* NSDC – Certification Course I & II applicable only from 2026 admitted batch onwards

R25 - Curriculum Structure
II Year I Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	BSC	25BAMATTC09	Probability and Statistics for Computer Science	3	0	0	3	3
2	ESC	25BACOMTC01	Computer Organization and Architecture	3	0	0	3	3
3	PCC	25BACAMTC02	Python Programming	3	0	0	3	3
4	PCC	25BACAMTC03	Algorithm Design and Complexity Analysis	3	0	0	3	3
5	PCC	25BACAMTC04	Operating System Concepts	3	0	0	3	3
6	PCC	25BACAMLC01	Python Programming Laboratory	0	0	2	2	1
7	PCC	25BACAMLC02	Algorithm Design and Complexity Analysis Laboratory	0	0	2	2	1
8	PCC	25BACAMLC03	Operating System Concepts Laboratory	0	0	2	2	1
9	SEC		**Skill Enhancement Course – I** *(Refer ANNEXURE - IV)*	1	0	2	3	2
10	EEC		**NSDC - Certification Course – III** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**16**	**0**	**8**	**24**	**20**

II Year II Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	BSC	25BAMATTC12	Discrete Mathematical Structures	3	0	0	3	3
2	PCC	25BACAMTC05	Object Oriented Programming Concepts	3	0	0	3	3
3	PCC	25BACAMTC06	Machine Learning Techniques	3	0	0	3	3
4	PCC	25BACAMTC07	Database Management and AI Integration	3	0	0	3	3
5	OE		**Open Elective – I** *(Refer ANNEXURE - II)*	3	0	0	3	3
6	PCC	25BACAMLC04	Object-Oriented Programming Concepts Laboratory	0	0	2	2	1
7	PCC	25BACAMLC05	Machine Learning Techniques Laboratory	0	0	2	2	1
8	PCC	25BACAMLC06	Database Management and AI Integration Laboratory	0	0	2	2	1
9	SEC		**Skill Enhancement Course – II** *(Refer ANNEXURE - IV)*	1	0	2	3	2
10	EEC		**NSDC - Certification Course – IV** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**16**	**0**	**10**	**26**	**20**

(L = Lecture, T = Tutorial, P = Practical, C = Credit)

R25 – Tentative Curriculum Structure
III Year I Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	HSMC		Management, Entrepreneurship and IPR	3	0	0	3	3
2	PCC	25BACAMTC08	Computer Networks	3	0	0	3	3
3	PCC	25BACAMTC09	Deep Learning	3	0	0	3	3
4	DE		**Discipline Elective – I** *(Refer ANNEXURE - III)*	3	0	0	3	3
5	OE		**Open Elective – II** *(Refer ANNEXURE - II)*	3	0	0	3	3
6	PCC	25BACAMLC07	Computer Networks Laboratory	0	0	2	2	1
7	PCC	25BACAMLC08	Deep Learning Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – III** *(Refer ANNEXURE - IV)*	1	0	2	3	2
9	INT	25BACAMIC01	Summer Internship - I	0	0	2	2	1
10	EEC		**NSDC - Certification Course – V** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**16**	**0**	**10**	**26**	**20**

III Year II Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	HSMC		Gender, Diversity and Inclusion	3	0	0	3	3
2	PCC	25BACAMTC10	Shell Programming	2	0	0	2	2
3	PCC	25BACAMTC11	Computer Vision and Image Processing	3	0	0	3	3
4	PCC	25BACAMTC12	IoT and Cloud Computing	2	0	0	2	2
5	DE		**Discipline Elective – II** *(Refer ANNEXURE - III)*	3	0	0	3	3
6	OE		**Open Elective – III (MOOC's)** *(Refer ANNEXURE - II)*	3	0	0	3	3
7	PCC	25BACAMLC09	Computer Vision and Image Processing Laboratory	0	0	2	2	1
8	PCC	25BACAMLC10	IoT and Cloud Computing Laboratory	0	0	2	2	1
9	AEC	25BAENGEC01	Soft Skills (Career Essentials)	1	0	2	3	2
10	EEC		**NSDC - Certification Course – VI** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**17**	**0**	**8**	**25**	**20**

(L = Lecture, T = Tutorial, P = Practical, C = Credit)

R25 - Tentative Curriculum Structure
IV Year I Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	PCC	25BACAMTC13	Natural Language Processing and Linguistics	3	0	0	3	3
2	PCC	25BACAMTC14	Generative AI	3	0	0	3	3
3	DE		**Discipline Elective – III** *(Refer ANNEXURE - III)*	3	0	0	3	3
4	DE		**Discipline Elective – IV** *(Refer ANNEXURE - III)*	3	0	0	3	3
5	DE		**Discipline Elective – V** *(Refer ANNEXURE - III)*	3	0	0	3	3
6	OE		**Open Elective – IV** *(Refer ANNEXURE - II)*	3	0	0	3	3
7	PCC	25BACAMLC11	Generative AI Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – IV** *(Refer ANNEXURE – IV)*	1	0	2	3	2
9	INT	25BACAMIC02	Summer Internship - II	0	0	4	4	2
10	EEC		**NSDC - Certification Course – VII** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**19**	**0**	**10**	**29**	**23**

IV Year II Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	PW	25BACAMPC01	Capstone Project	0	0	28	28	14
2	INT	25BACAMIC03	Full Semester Internship	0	0	6	6	3
Total				**0**	**0**	**34**	**34**	**17**

(L = Lecture, T = Tutorial, P = Practical, C = Credit)

ANNEXURE - I

THREE WEEK MANDATORY INDUCTION PROGRAMME

- Yoga and Meditation
- Sports and Games
- NSS
- NCC
- MITS Social Responsibility Club
- Management module
- Design Thinking
- Spoken and Written Communication

- ***Proficiency modules***

  - Basic Computer Proficiency
  - Interpersonal skills
  - Computer Graphics
  - Web programming
  - Mobile Apps
  - Vocabulary enhancement

HOLISTIC DEVELOPMENT ACTIVITIES

Description of Activities

1. Universal Human Values (UHV – I)
2. Physical and Health
3. Culture
4. Literature and Media
5. Social Service
6. Self-Development
7. Nature and Environment
8. Innovation

ANNEXURE - II

OPEN ELECTIVE – I (To be offered under MOOC's Category from SWAYAM – NPTEL)			
S. No.	**Course Code**	**Course Title**	**Course Offered by Department of**
1	25BAHUMOM01	English Language for Competitive Exams	Humanities and Social Sciences
2	25BAHUMOM02	Public Speaking	Humanities and Social Sciences
3	25BAHUMOM03	Indian Business History	Humanities and Social Sciences
4	25BAHUMOM04	Indian Economy: Some Contemporary Perspectives	Humanities and Social Sciences
5	25BAMBAOM01	E – Business	Management
6	25BAMBAOM02	AI in Human Resource Management	Management
7	25BAMBAOM03	AI in Marketing	Management
8	25BAMBAOM04	Artificial Intelligence for Investments	Management
9	25BACIVOM01	Plastic Waste Management	Civil
10	25BACIVOM02	Safety in Construction	Civil
11	25BAMECOM01	Operations Management	Mechanical
12	25BAEEEOM01	Transducers For Instrumentation	EEE
13	25BAECEOM01	Microprocessors and Interfacing	ECE
14	25BAECEOM02	Microprocessors and Microcontrollers	ECE
15	25BACOMOM01	Research Methodology	Multidisciplinary
Any new Interdisciplinary Course offered by SWAYAM NPTEL can be appended in future.			

ANNEXURE – III

LIST OF DISCIPLINE ELECTIVES

Discipline Elective – I (To be offered under MOOC's Category from SWAYAM – NPTEL)		
S. No.	**Course Code**	**Course Title**
1.	25BACAMDM01	Data Mining
2.	25BACAMDM02	Business Analytics and Text Mining Modeling using Python
3.	25BACAMDM03	Introduction to Soft Computing
4.	25BACAMDM04	Business Intelligence and Analytics
5.	25BACAMDM05	Cryptography and Network Security
6.	25BACAMDM06	Embedded Systems Design
Any advanced Course offered by SWAYAM NPTEL can be appended in future.		

Discipline Elective – II		
S. No.	**Course Code**	**Course Title**
1.	25BACAMDC01	Automata and Compiler Design
2.	25BACAMDC02	Graph Neural Networks and Applications
3.	25BACAMDC03	Predictive Analytics
4.	25BACAMDC04	Blockchain for AI
5.	25BACAMDC05	Reinforcement Learning
Any advanced courses can be appended in future.		

Discipline Elective – III		
S. No.	**Course Code**	**Course Title**
1.	25BACAMDC06	Web and Mobile Application development
2.	25BACAMDC07	Big Data Management and Analytics
3.	25BACAMDC08	Explainable AI and Model Interpretability
4.	25BACAMDC09	AI Driven Software Engineering and DevOps
5.	25BACAMDC10	Cyber Security and AI Driven Threat Detection
Any advanced courses can be appended in future.		

Discipline Elective – IV		
S. No.	**Course Code**	**Course Title**
1.	25BACAMDC11	AI for Finance
2.	25BACAMDC12	Multimodal Artificial Intelligence
3.	25BACAMDC13	MLOps and AI Model Deployment
4.	25BACAMDC14	Speech Systems
5.	25BACAMDC15	AI in Healthcare Informatics
Any advanced courses can be appended in future.		

Discipline Elective –V		
S. No.	**Course Code**	**Course Title**
1	25BACAMDC16	AI in LegalTech
2	25BACAMDC17	AI in Business Management and Analytics
3	25BACAMDC18	AI in AR/VR
4	25BACAMDC19	AI for Smart Cities and IoT Systems
5	25BACAMDC20	AI for Robotics
Any advanced courses can be appended in future.		

ANNEXURE – IV

List of Skill Enhancement Courses

Skill Enhancement Course – I		
S. No.	**Course Code**	**Course Title**
1.	25BACAMSC01	Data Science for Engineers
Any Courses can be appended in future.		

Skill Enhancement Course – II		
S. No.	**Course Code**	**Course Title**
1.	25BACAMSC02	Full Stack Development
Any Courses can be appended in future.		

Skill Enhancement course – III		
Sl. No.	**Course Code**	**Course Title**
1.	25BACAMSC03	Artificial Intelligence and Machine Learning Tools
2.	25BACAMSC04	Prompt Engineering
Any Courses can be appended in future.		

Skill Enhancement Course – IV		
Sl. No.	**Course Code**	**Course Title**
1.	25BACAMSC05	ML Optimization Techniques
Any Courses can be appended in future.		

ANNEXURE – V

NSDC Certification Courses

Program Track			
Cyber Security Specialization			
S. No.	**Course Code**	**Course Title**	**Certified By**
1	25BACOMCC01	Device Configuration and Management	Pearson VUE-ITS
2	25BACOMCC02	Networking (CISCO - CCST)	CISCO - CCST
3	25BACOMCC03	Network Security	Pearson VUE - ITS
4	25BACOMCC04	Microsoft Azure (AZ-900)	Microsoft
5	25BACOMCC05	Microsoft Security, Compliance, and Identity Fundamentals (SC-900)	Microsoft
6	25BACOMCC06	Cyber Security	CISCO - CCST
AIML Specialization			
7	25BACOMCC07	HTML 5 Application Development	Pearson VUE
8	25BACOMCC08	Python	Pearson VUE
9	25BACOMCC09	Artificial Intelligence (Pearson VUE)	Pearson VUE
10	25BACOMCC10	Artificial Intelligence (Intel)	Intel
11	25BACOMCC11	Machine Learning (Intel)	Intel
12	25BACOMCC12	Machine Learning (IBM)	IBM
13	25BACOMCC13	Deep Learning with Tensor Flow	Intel
14	25BACOMCC14	Azure AI (AI-901)	Microsoft
Generative AI Specialization			
15	25BACOMCC08	Python	Pearson VUE
16	25BACOMCC15	Data Visualization with Python	IBM
17	25BACOMCC10	Artificial Intelligence (Intel)	Intel
18	25BACOMCC11	Machine Learning (Intel)	Intel
19	25BACOMCC14	Azure AI (AI-901)	Microsoft
20	25BACOMCC16	Generative AI	CCS

Data Science			
S. No.	**Course Code**	**Course Title**	**Certified By**
21	25BACOMCC08	Python	Pearson VUE
22	25BACOMCC17	Statistics 101	IBM
23	25BACOMCC18	Data Analysis with Python	IBM
24	25BACOMCC15	Data Visualization with Python	IBM
25	25BACOMCC11	Machine Learning (Intel)	Intel
26	25BACOMCC14	Azure AI (AI-901)	Microsoft
Cloud Computing & Cybersecurity Specialization			
27	25BACOMCC02	Networking (CISCO-CCST)	CISCO-CCST
28	25BACOMCC01	Device Configuration and Management	Pearson VUE
29	25BACOMCC19	Cloud Computing	Pearson VUE
30	25BACOMCC14	Azure AI (AI-901)	Microsoft
31	25BACOMCC05	Microsoft security, compliance, and identity-(SC-900)	Microsoft
32	25BACOMCC20	Introduction to Containers, Kubernetes, and OpenShift	IBM
IOS Mobile Application Development Specialization			
33	25BACOMCC21	HTML and CSS	Pearson VUE
34	25BACOMCC22	JavaScript	Pearson VUE
35	25BACOMCC23	App Development with Swift Certified User	Apple-SWIFT
36	25BACOMCC24	App Development with Swift Associate	Apple-SWIFT
37	25BACOMCC25	Microsoft Power Platform Fundamentals (PL-900)	Microsoft
38	25BACOMCC26	DevOps	IBM

Data Analytics with Python Specialization			
S. No.	**Course Code**	**Course Title**	**Certified By**
39	25BACOMCC07	HTML5 Application Development	Pearson VUE
40	25BACOMCC08	Python	Pearson VUE
41	25BACOMCC17	Statistics 101	IBM
42	25BACOMCC27	Data Analytics	Pearson VUE
43	25BACOMCC25	Microsoft Power Platform Fundamentals (PL-900)	Microsoft
44	25BACOMCC28	Azure Data Fundamentals (DP-900)	Microsoft
Data Analytics with R Specialization			
45	25BACOMCC07	HTML5 Application Development	Pearson VUE
46	25BACOMCC29	R for Data Science	IBM
47	25BACOMCC30	Data Visualization with R	IBM
48	25BACOMCC31	Machine Learning with R	IBM
49	25BACOMCC10	Artificial Intelligence (Intel)	Intel
50	25BACOMCC28	Azure Data Fundamentals (DP-900)	Microsoft
Creative Design & Multimedia Specialization			
51	25BACOMCC07	HTML5 Application Development	Pearson VUE
52	25BACOMCC32	Adobe Photoshop	Adobe
53	25BACOMCC33	Adobe Illustrator	Adobe
54	25BACOMCC34	Adobe Animate	Adobe
55	25BACOMCC35	Autodesk Maya	Autodesk
56	25BACOMCC36	Adobe Premiere Pro	Adobe

IoT			
S. No.	**Course Code**	**Course Title**	**Certified By**
57	25BACOMCC07	HTML5 Application Development	Pearson VUE
58	25BACOMCC08	Python	Pearson VUE
59	25BACOMCC02	Networking (CISCO-CCST)	CISCO-CCST
60	25BACOMCC37	Internet of Things	IBM
61	25BACOMCC38	Azure Fundamentals (AZ-900)	Microsoft
62	25BACOMCC09	Artificial Intelligence (Pearson VUE)	Pearson VUE
63	25BACOMCC10	Artificial Intelligence (Intel)	Intel
Java Full Stack Development Specialization			
64	25BACOMCC07	HTML5 Application Development	Pearson VUE
65	25BACOMCC39	JAVA	Pearson VUE
66	25BACOMCC40	NoSQL MongoDB	IBM
67	25BACOMCC41	Advanced Java (Servlets & JSP)	Ethnotech
68	25BACOMCC42	Spring Framework	IBM
69	25BACOMCC38	Azure Fundamentals (AZ-900)	Microsoft
Python Full Stack Development Specialization			
70	25BACOMCC07	HTML5 Application Development	Pearson VUE
71	25BACOMCC43	Software Development	Pearson VUE
72	25BACOMCC08	Python	Pearson VUE
73	25BACOMCC40	NoSQL MongoDB	IBM
74	25BACOMCC44	Advance Python (Django / Flask)	Ethnotech
75	25BACOMCC38	Azure Fundamentals (AZ-900)	Microsoft

I Year I Semester

B. Tech I Year I Semester

25BAMATTC02 CALCULUS FOR COMPUTER SCIENCE

L	T	P	C
3	0	0	3

Pre Requisites: Mathematics at Intermediate or Equivalent Level

Course Description:

The course introduces the concepts of single variable and multivariable calculus with the view of its applications in various engineering fields. It prepares the students to develop various methods of finding derivatives and integrals; understanding of concepts related to continuous functions and enrich their experience in critical analysis.

Course Objectives:

1. To illustrate various techniques of testing the convergence of infinite series.
2. To introduces the Fourier Series expansions and beta, gamma functions.
3. To acquire knowledge on fundamental principles of differential calculus and polar graphing.
4. To familiarize the knowledge of limit, continuity, partial derivatives, extreme values in multivariable functions.
5. To emphasize the role of double and triple integrals in dealing with area and volume of the regions.

UNIT I SEQUENCES AND SERIES 9 hours

Sequences and Infinite Series, Convergence and tests for convergence (Comparison test, Ratio test, Root test, Leibniz's Test); Power series and convergence.

UNIT II FOURIER SERIES AND BETA-GAMMA FUNCTIONS 9 hours

Fourier Series, Euler's Formulae, Even and Odd functions, Cosine and Sine Series, Beta and Gamma functions.

UNIT III DIFFERENTIAL CALCULUS & POLAR GRAPHING 9 hours

Rolle's Theorem, Mean value theorems, Indeterminate forms of Limits, Taylor and Maclaurin series, Polar coordinates, Polar Graphing.

UNIT IV MULTIVARIABLE DIFFERENTIAL CALCULUS 9 hours

Functions of severable variables, Limits, Continuity, Partial derivatives, Chain rule, Directional derivative and gradient vectors, Extreme values and Saddle points, Constrained maxima and minima, Lagrange multipliers.

UNIT V MULTIVARIABLE INTEGRAL CALCULUS 9 hours

Multiple Integrals: Double integrals (Cartesian and polar), Reversing the order of integration (Cartesian), Change of integrals (Cartesian to polar), triple integrals, cylindrical and spherical coordinates, Jacobian, Substitutions in Multiple Integrals.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Use the Power series for ascertaining the stability and convergence of various techniques.

CO2: Apply Fourier series to represent periodic functions and evaluate integrals using Beta and Gamma functions.

CO3: Apply the mean value theorems, series expansions and tracing the polar curves in engineering.

CO4: Evaluate the rates of change in time and space variables through the analysis of multivariable functions in engineering.

CO5: Compute multiple integrals in various coordinate systems for engineering applications.

Text Books:

1. George B. Thomas, Maurice D. Weir, Joel R. Hass, Thomas' Calculus, Pearson Education 12th Edition, 2014.
2. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 42th Edition, 2012.

Reference Books:

1. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons, 9th Edition, 2006.
2. B.V. Ramana, Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
3. R.K. Jain, S.R.K. Iyengar, Advanced Engineering Mathematics, Alpha Science International Ltd. 4th Edition, 2014.
4. Michael D. Greenberg, Advanced Engineering Mathematics, Prentice Hall, 2nd Edition, 1998.
5. Stanley J. Miklavcic, An Illustrative Guide to Multivariable and Vector Calculus, Springer, 2019.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech I Year I Semester

25BACHETC03 SMART CHEMISTRY

L	T	P	C
3	0	0	3

Pre Requisites: Intermediate Chemistry

Course Description:

SMART Chemistry explores the intersection of modern chemistry, materials science, and computation. The course covers quantum chemistry, group theory, spectroscopy, nanomaterials, molecular logic gates, and bio-inspired computing, emphasizing their applications in molecular electronics and AI-driven chemical systems. Students gain a foundational understanding of advanced chemical principles and futuristic computing paradigms.

Course Objectives:

1. **To develop** an understanding of the principles of quantum chemistry, materials science, group theory, and spectroscopy relevant to computing and molecular technologies.
2. **To enable** students to apply concepts of materials for computing, molecular logic gates, and bio-inspired systems to solve problems in next-generation computing and electronics.
3. **To cultivate** analytical skills for interpreting chemical, spectroscopic, and computational data, and relating them to emerging scientific and technological applications.

UNIT I QUANTUM CHEMISTRY 9 hours

Fundamentals of Quantum mechanics:- Wave-particle duality, Uncertainty principle, Postulates of quantum mechanics, Schrodinger Wave equation, significance of Ψ and Ψ^2, particle in one dimensional box, molecular orbital theory – bonding in homo-diatomic molecules – energy level diagrams of N_2, and O_2, calculation of bond order, π-molecular orbitals of butadiene.

UNIT II GROUP THEORY & SPECTROSCOPY 9 hours

Introduction to group theory, definitions of group, subgroup, Class, Properties of a group, Symmetry elements and symmetry operations, Symmetry point groups, Matrix representation of groups, C_{2V} & C_{2h} point groups. Integration with Quantum Chemistry: Symmetry-informed AI models for molecular structure prediction.

Introduction to spectroscopy, UV-Visible spectroscopy – electronic transitions in MOs, Beer-Lambert's Law - Instrumentation – Applications.

UNIT III MATERIALS FOR COMPUTING 9 hours

Semiconductors – Introduction, basic concept, role of doping agents, applications

Super conductors-Introduction, basic concept, applications.

Conducting polymers – Introduction, mechanism of conduction & applications

Memristors: Concept, working principle, and applications

Nano materials: Introduction, classification, properties and applications (including computing) of Fullerenes, carbon nano tubes and Graphene.

UNIT IV MOLECULAR LOGIC GATES 9 hours

Introduction to Boolean logic, Basic logic gates: AND, OR, NOT, Truth tables and binary language, Transition from silicon to molecules: molecular electronics (VS) molecular logic, Simple chemical input–output systems (e.g., pH, metal ions), Fluorescent and colorimetric logic gates.
Basic examples: (AND gate using acid–base, OR gate using metal ions, NOT gate using quenching), Applications of Molecular Logic, Future of Molecular computing (Limitations and challenges of molecular logic, Comparison with traditional electronics.

Future of Molecular Computing: AI/ML-driven design of molecular circuits, hybrid silicon-molecular architectures.

UNIT V BIO-INSPIRED COMPUTING 9 hours

Introduction to biomolecules, structure & functions of carbohydrates, amino acids, protein, DNA, RNA, neurons, synapses, neuro-muscular junctions (NMJ).
Genes, codons, DNA computing, protein computing, biohybrid robots-Xenobots, Belousov–Zhabotinsky (BZ) reaction and applications.
Introduction to post-CMOS computing, Link to brain-inspired AI (neuromorphic computing)
AI in Bio-computing: ML for genome analysis, protein folding (AlphaFold).

Course Outcomes:

At the end of the course, the students would be able to

CO1: Explain the fundamental postulates of quantum mechanics and the significance of the wave function
CO2: Explain the concepts, properties, and applications of semiconductors, superconductors, conducting polymers, and memristors. Analyze semiconductors, superconductors, polymers, memristors, and nanomaterials for computing, with emphasis on AI-enabled materials discovery.
CO3: Explain Boolean logic, basic logic gates, and the transition from silicon to molecular electronics
CO4: Explain the structure and functions of biomolecules and their roles in biological computing systems
CO5: Explain the basic concepts of group theory, symmetry operations, and point groups, integrating AI/ML approaches for chemical and spectroscopic data interpretation.

Text Books:

1. Quantum Chemistry – Ira N. Levine, 8th Ed., Pearson Education, 2020.

2. Nanostructures and Nanomaterials: Synthesis, Properties and Applications – Guozhong Cao & Ying Wang, 2nd Ed., World Scientific, 2011.
Group Theory and Chemistry – David M. Bishop, Dover Publications, 1993.

3. Artificial Intelligence in Chemistry: Machine Learning and Deep Learning in Chemical and Materials Science – Edited by Hugh Cartwright, Royal Society of Chemistry, 2020.

4 Symmetry and Spectroscopy: An Introduction to Vibrational and Electronic Spectroscopy – Daniel C. Harris & Michael D. Bertolucci, Illustrated Reprint (Dover Publications), November 1, 1989.

5 Recent Developments in Biologically Inspired Computing – L. N. de Castro & F. J. Von Zuben, Idea Group Publishing, 2004.

6 Molecular Logic-based Computation – A. Prasanna de Silva, Royal Society of Chemistry, 2012

Reference Books:

1. Molecular Electronics: An Introduction to Theory and Experiment – Juan Carlos Cuevas & Elke Scheer, 2nd Ed., World Scientific, 2017.
2. Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies – Dario Floreano & Claudio Mattiussi, MIT Press, 2008.
3. Principles of Instrumental Analysis – Douglas A. Skoog, F. James Holler, and Stanley R. Crouch, 7th Ed., Cengage Learning, 2018.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B.Tech I Year I Semester

25BACIVTC02 INTRODUCTION TO ENVIRONMENT AND SUSTAINABILITY

		L	T	P	C
Pre-Requisites:	None	**2**	**0**	**0**	**2**

Course Description:

This course provides foundational knowledge of environmental systems, biodiversity, and sustainability. It addresses major environmental challenges like pollution and climate change, and introduces students to key policies, global sustainability frameworks, and sustainable engineering practices.

Course Objectives:

1. Understand the interdependence between environmental system and human well-being.
2. Analyze key environmental issues including pollution, climate change, and resource degradation.
3. Interpret environmental laws, standards, and global sustainability agreements.
4. Evaluate sustainability frameworks and the role of engineering in achieving development goals.
5. Apply sustainable engineering principles to address real-world environmental challenges.

UNIT I ENVIRONMENTAL SYSTEMS 6 hours

Definition, scope, and importance of environment - Components of ecosystem: biotic and abiotic - Energy flow in ecosystems (food chains, food webs) - Biogeochemical cycles: Water, Carbon, Nitrogen - Biodiversity: definition, values, and threats - Overview of natural resources: renewable vs. non-renewable

UNIT II ENVIRONMENTAL CHALLENGES 6 hours

Pollution: definition, types (Air, Water, Soil, Noise) - Causes, effects, and basic control measures - Urbanization – Land degradation - Solid and E-waste management –- Climate change, global warming, greenhouse effect - Ozone depletion and carbon footprint

UNIT III ENVIRONMENTAL POLICIES 6 hours

Overview of ISO 14001:2015 - Roles of CPCB, SPCBs, and NGT - Key environmental acts (Air, Water, Forest, Wildlife) - International agreements: Montreal, Kyoto, Paris Agreement - Basics of Environmental Impact Assessment (EIA) - Introduction to carbon credit and energy auditing

UNIT IV SUSTAINABILITY GOALS 6 hours

Foundations of Sustainability- Definition, Concept, needs and challenges- principles of sustainability - Evolution of Global Sustainability Frameworks - Agenda 21, millennium development goals, and protocols - Sustainable Development Goals - targets, indicators – Role of engineering and technology in achieving SDGs

UNIT V SUSTAINABLE ENGINEERING APPROACHES 6 hours

Introduction to Sustainability in Engineering - Guiding principles and Frameworks for sustainable engineering – Sustainability approaches - Triple bottom Line, Cradle to Cradle concept - Life cycle assessment - Zero waste and R concept - Circular economy - ISO 14000 Series

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Recommend ecofriendly solution to protect ecosystem.
CO2: Identify appropriate pollution control methods for specific situations.
CO3: Demonstrate the use of renewable energy for sustainable development.
CO4: Apply techniques to reduce environmental impact from climate change.
CO5: Utilize environmental laws and policies to promote sustainable practices.

Text Books:

1. R. L. Rag and Lekshmi Dinachandran Remesh. Introduction to Sustainable Engineering. 2nd Edition,PHI Learning Pvt. Ltd., 2016.
2. Singh, J.S., Singh, S.P. and Gupta, S.R. 2014. Ecology, Environmental Scienceand Conservation. S. Chand Publishing, New Delhi.
3. **Masters, Gilbert M. & Ela, Wendell P.** – *Introduction to Environmental Engineering and Science*, Pearson Education, 3rd Edition, 2013
4. Bradley. A.S; Adebayo,A.O., Maria, P. Engineering applications in sustainable design and development,Cengage learning
5. Allen, D. T. and Shonnard, D. R., Sustainability Engineering: Concepts, Design and Case Studies,Prentice Hall
6. Richard T. Wright, Dorothy F. Boorse (2017) Environmental Science: Toward A Sustainable Future,Pearson, 13th Edition

Reference Books:

1. R.C. Sohal & S.K. Agrawal – *Environmental Studies*, University Science Press, 2018.
2. Erach Bharucha – *Textbook of Environmental Studies for Undergraduate Courses*, University Grants Commission, 2nd Edition, 2013.
3. Peavy, H.S., Rowe, D.R. & Tchobanoglous, G. – *Environmental Engineering*, McGraw Hill Education, 2017.
4. R.K. Trivedi & P.K. Goel – *An Introduction to Air Pollution*, B.S. Publications, 2015.
5. Harris, Frances (2012) Global Environmental Issues, 2nd Edition. Wiley-Blackwel
6. Rajagopalan, R. (2011). Environmental Studies: From Crisis to Cure. India: Oxford University Press.
7. Leelakrishnan, P. (2022). *Environmental law in India* (Vol. 1). LexisNexis.
8. Ghosh, S. (Ed.). (2019). *Indian environmental law: Key concepts and principles*. Orient BlackSwan.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech I Year I Semester

25BAEEETC03 DIGITAL DESIGN

L	T	P	C
3	0	0	3

Pre Requisites: NIL

Course Description:

This course provides a modern introduction to logic design and the basic building blocks used in digital systems, in particular digital computers. It starts with a discussion of combinational logic, and also the course deals with sequential circuits, State machines, Different representations including truth table; logic gate, timing diagram, switch representation, and state diagram will be discussed.

Course Objectives:

1. To introduce number systems, logic gates, and Boolean algebra.
2. To provide methods for simplifying logical expressions and designing combinational circuits.
3. To develop understanding of sequential circuits and their applications.
4. To explain standard programmable logic devices and memory elements.
5. To familiarize students with digital system design practices.

UNIT I NUMBER SYSTEMS AND LOGIC GATES 9 hours

Number systems and conversions: binary, decimal, octal, hexadecimal. Signed and unsigned numbers. Binary arithmetic: addition, subtraction, multiplication. Logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR – truth tables and logic symbols. NAND and NOR Implementation

UNIT II BOOLEAN ALGEBRA AND MINIMIZATION 9 hours

Boolean postulates and laws. DeMorgan's theorems. Standard SOP and POS forms. Boolean Algebra, Minimization of Logic expressions, Quine–McCluskey Method/Tabulation Method, K-Map Simplification (Upto 4-Variable). Realization using logic gates.

UNIT III COMBINATIONAL LOGIC CIRCUITS 9 hours

Design and implementation of combinational circuits: half adder, full adder, subtractor, comparator, multiplexer, demultiplexer, encoder, decoder. Code Converters like Binary to Gray, Gray to Binary conversion.

UNIT IV SEQUENTIAL LOGIC CIRCUITS 9 hours

Latches and flip-flops: SR, JK, D, T – truth tables and excitation tables. Conversion between flip-flops. Counters: asynchronous and synchronous. Shift registers and basic timing diagrams.

UNIT V MEMORY AND PROGRAMMABLE LOGIC DEVICES 9 hours

Memory Hierarchy and classification of memories - RAM, ROM, PROM, EPROM different types. Error Detection and Correction, Read-only Memory, Programmable Logic Array, Programmable Array Logic.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Convert and perform arithmetic operations on different number systems.
CO2: Simplify Boolean expressions and implement logic functions.
CO3: Design combinational circuits using standard building blocks.
CO4: Analyze and implement sequential circuits using flip-flops and counters.
CO5: Explain the working of memory devices and programmable logic components.

Text Books:

1. Digital Design, M. Morris Mano, Micheal D. Ciletti, 5th Edition, 2013, Pearson.
2. Digital Logic Design, Leach, Malvino, Saha,TMH

Reference Books:

1. Donald D. Givonne, "Digital Principles and Design" TMH, 2003. Digital Logic & State Machine Design, David J. Comer, Oxford University Press, 3rd Reprinted Indian Edition, 2012.
2. Digital Logic Design, R.D. Sudhakar Samuel, Elsevier
3. Computer System Architecture , M. Morris Mano, 3th Edition, pearson.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech I Year I Semester

25BACSETC01 C PROGRAMMING FOR PROBLEM SOLVING

L	T	P	C
3	0	0	3

Pre-requisite: None

Course Description:
This course introduces algorithmic problem-solving and the fundamentals of C programming. It emphasizes flowcharts, pseudocode, and structured programming constructs, covering functions, arrays, pointers, strings, user-defined data types, and file handling to develop efficient computational solutions.

Course Objectives:
1. Understand the fundamental building blocks of algorithms such as statements, state, control flow, and functions.
2. Apply algorithmic strategies like iteration and recursion for problem solving.
3. Represent and analyze algorithms using flowcharts, pseudocode, and programming constructs.
4. Differentiate control structures and algorithm design techniques through practical problems.
5. Develop comprehensive programming solutions using C syntax and semantics (variables, loops, functions, arrays, pointers, file handling).

UNIT I ALGORITHMIC PROBLEM SOLVING **9 hours**

Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), strategies for developing algorithms. Introduction to scratch & raptor.

UNIT II INTRODUCTION TO C PROGRAMMING **9 hours**

Structure of C Program, C Tokens: Variables, Data types, Constants, Identifiers, key words and Operators, Expressions, Storage classes.
Control Structures: Conditional Statements, Looping Statements.

UNIT III FUNCTIONS AND ARRAYS **9 hours**

Introduction to Functions, Function Definition and Declaration, Function prototype, Function Parameters, Passing arguments and arrays to functions, Recursion.
Introduction to Arrays, Types of arrays and operations.

UNIT IV POINTERS AND STRINGS **9 hours**

Introduction to Pointers, dereferencing and address operators, pointer and address arithmetic, array manipulation using pointers .
Introduction to Strings: String Operations and String functions.

UNIT V USER DEFINED DATA TYPES & FILE HANDLING **9 hours**

User-defined data types: Structure and Union, Dynamic memory allocation.
Introduction to file concepts, File handling operations.

Course Outcomes:
CO1: Explain fundamental algorithmic concepts and represent problems using pseudocode and flowcharts.
CO2: Implement C programs using control structures such as conditionals and loops.
CO3: Design and use functions and arrays for modular programming.

CO4: Apply pointers, dynamic memory allocation, and string operations to manipulate data effectively.

CO5: Demonstrate file handling techniques and user-defined data types for structured programming solutions.

Text Books:

1. "The C Programming Language", Brian W. Kernighan and Dennis M. Ritchie, Prentice- Hall, 1988
2. C Programming, A Problem Solving Approach, Forouzan, Gilberg,Prasad, CENGAGE, 3rd edition.

Reference Books:

1. Programming in C, Rema Theraja, Oxford, 2016, 2nd edition
2. Schaum's Outline of Programming with C, Byron S Gottfried, McGraw-Hill Education, 1996

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech I Year I Semester

25BAMECEC01 ENGINEERING GRAPHICS

L	T	P	C
2	0	2	3

Pre-requisite: None

Course Description:
This course introduces the fundamentals of Engineering Graphics and its applications, with hands-on practice using AutoCAD and Autodesk Fusion 360. It covers orthographic projection of points, lines, planes, and solids, including sectional views and surface developments. Students will learn to interpret and convert between isometric and orthographic views, as well as perform simple 2D drawings and 3D modelling. The course emphasizes visualization skills and technical drawing practices essential for engineering design and communication.

Course Objectives:
This course is designed to:

1. Provide fundamental knowledge of engineering graphics and its applications in design and communication.
2. Develop skills in orthographic projections of points, lines, planes, and solids.
3. Enable students to visualize and represent sections and development of engineering solids.
4. Familiarize students with isometric and orthographic views and their interconversion.
5. Introduce computer-aided design (AutoCAD and Autodesk Fusion 360) for 2D drafting and simple 3D modelling.

UNIT I INTRODUCTION **12 hours**

Introduction to Engineering Graphics and its Applications. Introduction to AutoCAD commands. Types of Lines, Dimensioning and Geometrical Constructions, Simple 2D drawings using AutoCAD.

UNIT II PROJECTIONS OF POINTS & LINES **12 hours**

Projection of points: Orthographic projections, notation system, positions and projection of points in four quadrants.
Projection of lines: Positions and projection of lines (inclined to one plane, HP/VP).

UNIT III PROJECTIONS OF PLANES & SOLIDS **12 hours**

Projection of planes: Positions and projection of planes (inclined to two planes, HP and VP).
Projection of solids: Projections of regular solids. (resting on HP and axis inclined to HP only).

UNIT IV SECTION OF SOLIDS AND DEVELOPMENT OF SURFACES **12 hours**

Section of solids: Sectional view of regular solids (cutting plane inclined to HP) and its true shapes.
Development of Surfaces: Development of surfaces of regular solids (Prism and Pyramid only).

UNIT V ISOMETRIC, ORTHOGRAPHIC VIEWS AND SIMPLE 3D MODELLING **12 hours**

Conversion of isometric views into orthographic views and vice-versa.
3D Modelling: Introduction to 3D modelling. Simple 3D modelling using Autodesk Fusion 360. (Software practice only. Not for examination)

Course Outcomes:

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

CO1: Interpret and apply AutoCAD commands to create simple 2D engineering drawings.
CO2: Construct orthographic projections of points, lines, and planes in different quadrants.
CO3: Generate projections of solids, sectional views, and true shapes.
CO4: Develop surface diagrams of prisms and pyramids and apply them to practical applications.
CO5: Convert between isometric and orthographic views and create simple 3D CAD models using Fusion 360.

Text Books:

1. K.L. Narayana & P. Kannaiah, *Engineering Graphics*, 4th Edition, Scitech Publications, 2021.
2. K. Venugopal & V. Prabhu Raja, *Engineering Drawing + AutoCAD*, 7th Edition, New Age International, 2022.
3. Dhananjay A. Jolhe, *Engineering Drawing with an Introduction to AutoCAD*, 5th Edition, McGraw Hill Education, 2023.

Reference Books:

1. N.D. Bhatt & V.M. Panchal, *Engineering Drawing: Plane and Solid Geometry*, 56th Edition, Charotar Publishing House, 2023.
2. Shah, P.J., *Engineering Drawing*, 3rd Edition, Pearson Education, 2021.
3. Agrawal B. & Agrawal C.M., *Engineering Graphics*, 2nd Edition, Tata McGraw Hill, 2020.

Online Learning Resources:

1. National Programme on Technology Enhanced Learning (NPTEL), Engineering Graphics – IIT Kharagpur. Available at: https://nptel.ac.in/courses/11210560
2. Autodesk Education, AutoCAD & Fusion 360 Learning Resources. Available at: https://www.autodesk.com/education/edu-software/overview
3. Coursera, Engineering Drawing and Visualization – Georgia Tech. Available at: https://www.coursera.org/learn/engineering-drawing

Mode of Evaluation: Continuous Assessment, Mid Term Tests and End Semester Examination.

B. Tech I Year I Semester

25BACHELC03 CHEMISTRY LABORATORY

L	T	P	C
0	0	2	1

Pre Requisites: NIL

Course Description:

This laboratory course provides practical training in classical volumetric and instrumental analytical techniques, alongside modern experimental approaches including nanomaterials synthesis, conducting and biodegradable polymers, spectroscopy, and molecular logic gates. Emphasis is placed on sustainable methodologies, quantitative data analysis, and applications of bio-inspired computing in chemical systems.

Course Objectives:

- To verify the fundamental concepts with experiments

List of Experiments:

From the following 17 experiments, students are required to perform any 10 using volumetric and/or instrumental methods of analysis. Wherever applicable, modeling software may also be used.

1. Preparation of a nanomaterial.
2. Demonstration of a two-input molecular AND logic gate based on two chemical inputs using spectroscopy
3. Demonstration of a three-input molecular AND logic gate based on three chemical inputs
4. using spectroscopy
5. Determination of λmax for molecules using UV-VISIBLE spectroscopy
6. Synthesis of polyaniline conducting polymer
7. Demonstration of a two-input molecular OR logic gate based on two chemical inputs
8. Estimation of ferrous ion by colorimetry
9. Demonstration of a two-input molecular AND logic gate based on two chemical inputs using titrimetry
10. Green synthesis of PVA/Starch biopolymer (biodegradable materials for sustainable tech)
11. Synthesis of PVA biopolymer
12. Estimation of Ascorbic acid
13. Demonstration of a two-input molecular NOT logic gate based on two chemical inputs
14. Verification of Beer-Lambert's Law
15. Determine reaction order and rate constant using MS-excel regression model

16. Conductometric titration of strong acid vs strong base
17. Conductometric titration of weak acid vs strong base
18. Determination of amino-acid sequence and ID of a protein from a DNA/protein input.

Course Outcomes:

CO1: Apply volumetric, colorimetric, conductometric, and spectroscopic techniques to determine physicochemical properties and composition of chemical systems.
CO2: Demonstrate the principles of molecular logic gates and their applications in bio-inspired computing through chemical and spectroscopic experiments.
CO3: Design and synthesize nanomaterials, conducting polymers, and biodegradable polymers, highlighting sustainable and green chemistry approaches.
CO4: Analyze kinetic parameters and quantitative data using modern computational tools such as regression models and molecular modeling software.
CO5: Integrate experimental chemistry with interdisciplinary applications in materials science, biopolymers, and biomolecular identification for real-world problem solving.

Text / Reference Books:

1. Vogel's Textbook of Quantitative Chemical Analysis – J. Mendham, R.C. Denney, J.D. Barnes & M.J.K. Thomas, 6th Ed., Pearson Education, 2000.
2. Instrumental Methods of Chemical Analysis – B.K. Sharma, 23rd Ed., Goel Publishing House, 2007.
3. Molecular and Supramolecular Information Processing: From Molecular Switches to Logic Systems – Evgeny Katz (Ed.), Wiley-VCH, 2012.

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech I Year I Semester

25BACSELC01 C PROGRAMMING FOR PROBLEM SOLVING LABORARTORY

L	T	P	C
0	0	2	1

Pre-requisite: NIL

Course Description:

This course introduces the fundamentals of problem-solving and algorithmic thinking, covering flowcharts, pseudocode, and structured programming in C. Students will learn to apply control structures, arrays, functions, pointers, strings, and dynamic memory in developing efficient solutions. Practical exercises include recursion, file handling, and real-world problem-solving through structured programming projects.

Course Objectives:

1. Understand the principles of problem analysis and algorithm design for computational tasks.
2. Represent solutions using flowcharts and pseudocode.
3. Apply control structures, arrays, functions, and recursion in solving problems.
4. Develop programs using pointers, strings, structures, and dynamic memory allocation.
5. Implement file handling and practice real-world problem-solving through C programming.

List of Experiments

1. Develop a flowchart using Raptor for the following tasks

a) Develop a flowchart for the various arithmetic operations on numbers.
b) Develop a flowchart to check whether the number is positive or negative.
c) Develop a flowchart for finding biggest number among three numbers.
d) Develop a flowchart to print factorial of a number using function.
e) Develop a flowchart to check given number is palindrome or not using function.

2. A climate research center stores temperature readings for two different cities in the wrong order.

First, swap the average temperature values of the two cities using:
1.A temporary variable
2.Without a temporary variable (using arithmetic or bitwise operations).
Then, if the swapped temperatures are stored in variables t1 and t2, determine which city is hotter and calculate the absolute difference in their average temperatures.

3. Admission Eligibility & Scientific Calculator
a) A college is developing admission software for a professional course. Based on a student's marks in Maths, Physics, and Chemistry:
If the student has Maths ≥ 65, Physics ≥ 55, and Chemistry ≥ 50 OR the total in all three subjects ≥ 180, they're eligible; otherwise, they're not.
Write a program to automate this check.
b)Your friend is tired of using a mobile calculator and wants a simple scientific calculator on their computer that can perform addition, subtraction, multiplication, division, and other advanced mathematical operations.

4. Multiplication Table, Range Sum & Prime Numbers
a)Your cousin just started learning multiplication and asks you to create a program that prints the

multiplication table for any number they enter.
b)An online form accepts numbers between 0 and 100. The form keeps storing these numbers until you enter -1. You need to find the sum of all valid numbers entered before -1.
c) Your mathematics professor wants a program that lists all prime numbers between two given limits for research purposes.

5. Matrix Operations
You are working on an image-processing system where images are stored as square matrices (N×N). The system must:
• Add two matrices (combine pixel intensities)
• Multiply two matrices (apply transformation)
Write a program to support both operations.

6. String Operations without Libraries
(a) A company wants a secure text-processing library without using standard string functions (to avoid hidden bugs). Implement:
• String length calculation
• String copying
• String comparison
• String concatenation
• String reverse
all without using <string.h> functions.

(b) You are working as a software developer at a customer support centre. The centre receives numerous customer feedback messages daily, stored as strings. The manager wants a program that can help efficiently analyse these messages.
Task:
Write a C program that takes an array of customer feedback strings and a keyword, then returns the count of feedback entries that contain the given keyword (case-insensitive).
Key points to consider:

- Input: An array of strings representing customer feedback and a keyword string.
- Output: The number of feedback messages that contain the keyword.
- The search should be case-insensitive.
- Efficient string searching or matching techniques should be applied.
- Handle scenarios where the keyword may appear multiple times within the same feedback.

7. Array Sorting with Pointers & Dynamic Memory Allocation
a) A small inventory system stores a list of product prices. The manager wants them sorted using pointer-based operations instead of the usual array index access.
b) You are part of a community sports club that is organizing a local athletics meet. Participants will compete in various running events, and each participant's *name* and *completion time in seconds* will be recorded after the race.
The number of participants is not known beforehand, as registrations happen until the race starts.
Task:

Write a C program that:

1. Dynamically allocates memory to store the race results for N participants (entered at runtime).
2. Stores each participant's name and their completion time.
3. Sorts the participants in ascending order of completion time (fastest first) using pointer-based

array manipulation.
4. Displays the sorted results with ranking (1st fastest, 2nd fastest, etc.).

8. Time Difference Calculation
(a) A transport company records the departure and arrival times of buses (in hours, minutes, and seconds). You need to find the travel time between the two times using a structure.
(b) A swimming pool management system keeps track of customers' swimming session times. For each customer, the start time and end time of their swimming session are recorded (in hours, minutes, and seconds). The system needs to calculate the total time spent in the pool for each customer.
Task:
Write a C program that:
1. Uses a structure to store time in hours, minutes, and seconds.
2. Reads the start time and end time of the swimming session.

9. Recursion & Passing Parameters
a) Your mathematics professor assigns you a program to find factorial, greatest common divisor (GCD), and Fibonacci numbers both with recursion.
b) You are working as a developer for a fitness tracking system. The system needs to update a user's daily step count. The update can be done in two ways:
- Using pass-by-value where the function receives a copy of the step count and cannot modify the original value.
- Using pass-by-reference where the function receives a pointer to the step count and can directly update the original value.

10. File Operations
You are working at a library to manage the record of books issued to members. The library maintains a file that stores details of issued books such as book ID, member ID, and the issue date. Write a C program that performs the following operations using file handling:
1. Add new records: Allow the librarian to add multiple book issue records to a file.
2. Display all records: Read and display all the records stored in the file.
3. Search a record: Search for a particular book issue record by book ID and display the details if found.

Course Outcomes:

After completion of the course, Students will be able to

CO1: Implement algorithms and debug simple programs to enhance problem-solving skills.

CO2: Apply conditional statements and loops for controlling program execution.

CO3: Solve real-world tasks using arrays, strings, and related programming constructs.

CO4: Create and use functions, apply pointers, and understand parameter passing techniques.

CO5: Implement structures, dynamic memory allocation, and file handling for practical applications.

Text / Reference Books:

1. Ajay Mittal, Programming in C: A practical approach, Pearson.
2. C Programming, A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE
3. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice- Hall of India
4. Byron Gottfried, Schaum's Outline of Programming with C, McGraw Hill

Mode of Evaluation: Continuous Internal Evaluation, Model Test and End Semester Examination

B. Tech I Year I Semester

25BACOMLC01 ENGINEERING SKILLS LABORATORY

L	T	P	C
0	0	2	1

Pre-Requisites Nil

Course Description:

This course provides hands-on experience in using basic electrical/electronic instruments and circuit design. and to develop practical skills in microcontroller programming and device interfacing for real-time applications. And provides hands-on training in fundamental manufacturing and fabrication processes used in mechanical engineering. Students will develop practical skills through activities such as sheet metal product development, plastic fabrication, 3D printing, woodwork using carpentry tools, laser engraving, and metal welding. The course also introduces the operation and applications of various mechanical power tools, CNC lathe, and milling machines, enabling learners to gain exposure to modern and traditional manufacturing techniques.

Course Objectives:

By the end of this course, the student will be able to:

1. To provide hands-on experience in using basic electrical/electronic instruments and circuit design.
2. To develop practical skills in microcontroller programming and device interfacing for real-time applications.
3. Provide hands-on experience in basic fabrication processes using sheet metal, plastics, wood, and welding to develop simple utility products.
4. Familiarize students with modern manufacturing technologies such as 3D printing, CNC machining, and laser engraving for product realization.
5. Enable students to understand, identify, and safely operate various mechanical engineering tools and equipment for engineering applications.

PART – A

Study Experiments:

1. Familiarization with Instrumentation and Tools
2. Understanding the microcontroller architecture and Programming

Choose any six from the following list of experiments:

1. Clock Pulse Generation and Signal Analysis using Digital Storage Oscilloscope (DSO).
2. Measurement of Electrical Power, Power Factor, and Energy using Portable Meters.
3. Study, Design and Testing of Full-Wave Bridge Rectifier Circuit.
4. Study, Design and Implementation of a SMPS.
5. Study, Design and Testing of Multiplexer using Logic Gates.
6. Interfacing of 7-Segment Display, Buzzer, Sensor with microcontroller.
7. Speed and Angle Control of Servo Motor using Microcontroller.
8. PCB Soldering Techniques and Testing Using a Multimeter.

Content Beyond the Syllabus (Virtual Laboratory)

9. Familiarisation of ICs.
10. Application of basic logic gates in fire and burglar alarms.
11. Wiring of a simple circuit for controlling

    (1) a lamp/fan point,
    (2) a staircase or corridor winding,
    (3) an electrical appliance (16A socket).

PART – B

List of Experiments:

1. Utility product development using sheet metal.
2. Plastic product fabrication.
3. 3D printing of simple components.
4. Wood product fabrication using carpentry tools.
5. LASER engraving.
6. Metal welding and fabrication.
7. Study of different mechanical engineering power tools.
8. Study of CNC Lathe and Milling machining processes.

Content beyond Syllabus

9. Plumbing
10. Computer Controlled Cutting of wooden object
11. 3D Machining

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

CO1: Demonstrate soldering, measurement, and testing techniques using basic electrical/electronic instruments.

CO2: Design and implement simple circuits and microcontroller-based applications for power and control systems

CO3: Demonstrate the ability to fabricate simple components using sheet metal, wood, and welding processes.

CO4: Analyze and compare traditional and modern manufacturing techniques such as 3D printing, CNC machining, and laser engraving.

CO5: Select and operate appropriate hand tools, power tools, and machine tools for given product development tasks with safety considerations.

Text Books:

1. R. S. Sedha, A Textbook of Applied Electronics, S. Chand & Company Ltd.
2. Muhammad Ali Mazidi, The 8051 Microcontroller and Embedded Systems, Pearson Education.
3. V. Ramesh Babu, Engineering Workshop practice for JNTU, VRB Publishers Pvt. Ltd. 2023.
4. A. K. Sarathe, Engineering Workshop Practice, 1st edition, Khanna Book Publishers, 2022
5. Lab manual provided by the departments.

Reference Books:

1. Boylestad & Nashelsky, Electronic Devices and Circuit Theory, Pearson Education.
2. David A. Bell, Electronic Instrumentation and Measurements, Oxford University Press.
3. The Art of Electronics by Paul Horowitz and Winfield Hill, Cambridge University Press.
4. Sanjay Gupta & Santosh Gupta, SMPS: Switch Mode Power Supply, Technical Publications.
5. Boylestad & Nashelsky, Electronic Devices and Circuit Theory, Pearson Education.
6. P.Kannaiah, K.L.Narayana, Workshop Manual, 2nd Edition, SciTech Publishers, 2009.
7. K.C. John, Mechanical Workshop Practice, 2nd edition, Kindle Edition, 2010.
8. https://fab-coep.vlabs.ac.in/

Mode of Evaluation: Continuous Internal Evaluation, Model Test and End Semester Examination.

I Year II Semester

B. Tech I Year II Semester

25BAENGTC01 PROFESSIONAL COMMUNICATION

L	T	P	C
2	0	0	2

Pre Requisites: Universal Human Values (UHV-I) (desirable).

Course Description :

This course aims to develop essential English communication skills required for academic, social, and professional contexts. Students will learn to convey information effectively. The program enhances reading for comprehension, promotes reading for pleasure, and trains students to write various texts including emails, reports, business letters, presentations, and posters, enabling clear and confident communication in diverse settings.

Course Objectives:

This course aims to:

1. Enable students greet, introduce themselves and others, and describe their daily routines, surroundings, and familiar places.
2. Develop the ability to narrate past events or incidents coherently and to express plans and predictions for the future.
3. Foster functional communication skills such as enquiring, requesting, giving directions, instructions, and reporting information accurately.
4. Cultivate reading proficiency by engaging with stories, enhancing comprehension, vocabulary, and critical appreciation.
5. Equip students with skills in professional writing through emails, reports, letters, posters, and visual presentations.

UNIT I EVERYDAY ENGLISH 6 hours

Basics of essential grammar; Functions of communication such as greetings, introductions, leave-taking, polite expressions; describing daily routines and habits using the simple present tense; Describing things related to family, friends, classroom, home, campus, common places, gadgets and other objects.

UNIT II NARRATION AND PLANNING 6 hours

Using past tense forms for narration; sequencing events using connectors; vocabulary related to experiences, festivals, travel, and memorable events; expressing future time; predicting events and making career plans/prospects

UNIT III FUNCTIONAL COMMUNICATION IN DAILY LIFE 6 hours

Making enquiries using wh-questions, polite questions, and indirect requests; requesting and offering help with expressions for permission, obligation, and necessity; giving directions using common roadmap vocabulary; providing and following formal/informal instructions in everyday and academic contexts; reporting information through direct and indirect speech; impersonal passive voice.

UNIT IV READING FOR COMPREHENSION AND PLEASURE 6 hours

Developing extensive reading skills through prescribed short stories: *The Victory* by Rabindranath Tagore and *The Ransom of the Red Chief* by O Henry; applying techniques of reading such as skimming, scanning, and reading for gist and detail; building vocabulary through the texts; engaging in discussions on characters, settings, and themes from the reading; practicing reading comprehension for specific details through formal email samples.

UNIT V PROFESSIONAL AND BUSINESS COMMUNICATION 6 hours

Learning principles of professional communication such as clarity, conciseness, courtesy, correctness, and formats; Email etiquette; Drafting emails/letters critically appreciating the novel/short story read in Unit 4; preparing short reports; drafting business letters such as inquiries, complaints, replies; designing posters with focus on layout, persuasive language, and visuals; delivering short presentations using visual aids.

Course Outcomes:

Students will be able to:

CO1: Greet and introduce themselves and others, describe their routines, places, and things around them.

CO2: Narrate an event or incident, planning and predicting the future

CO3: Enquire, request, give directions/instructions, report and convey information

CO4: Read for pleasure and read for comprehension

CO5: Write formal/informal emails, short reports, presentations, business letters, and make posters.

Text Books:

1. Functional English for Communication by Ujjwala Kakarla, Tanu Gupta, Leena Pundir (SAGE, 2019)
2. Communication Skills by Sanjay Kumar & Pushp Lata (Oxford University Press, 3rd Edition, 2024)
3. Extensive Reading in the Second Language Classroom" by Richard R. Day and Julian Bamford (Cambridge University Press, 2022).

Reference Books:

1. Functional English Grammar (Cambridge University Press, 2024)
2. Essentials of Business Communication by Mary Ellen Guffey & Dana Loewy (Cengage Learning, 10th Edition, 2016)
3. English for Business Communication by Mable Chan (Routledge, 2025)
4. Handbook of Communication Skills edited by Owen Hargie (Routledge)

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination

B.Tech I Year II Semester

25BAHUMTC01 HUMAN VALUES AND PROFESSIONAL ETHICS

L	T	P	C
2	0	0	2

Pre-requisite None. Universal Human Values (UHV-I) (desirable).

Course Description :

1. This course presents a universal approach to value education by developing the right understanding of reality (i.e., a worldview of the reality "as it is") through the process of self-exploration.
2. The whole course is presented in the form of a dialogue whereby a set of proposals about various aspects of reality are presented, and the students are encouraged to self-explore the proposals by verifying them based on their natural acceptance within oneself and validating experientially in living.
3. The prime focus throughout the course is toward affecting a qualitative transformation in the life of the student rather than just a transfer of information.
4. While introducing the holistic worldview and its implications, a critical appraisal of the prevailing notions is also made to enable the students to discern the difference on their own right.
5. Thus, this course is intended to provide a much-needed orientation input in value education to the young enquiring minds.

Course Objectives :

1. To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity, which are the core aspirations of all human beings.
2. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a right understanding of the Human reality and the rest of Existence. Such a holistic perspective forms the basis of Universal Human Values (UHV) and movement towards value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behaviour and mutually enriching interaction with Nature.
4. To aid the students in realising their full human potential and act accordingly.
5. To assist the students to live with a feeling of relationship, harmony and co-existence.

UNIT I INTRODUCTION TO VALUE EDUCATION 6 hours

Lecture 1: Holistic Development and the Role of Education (Right Understanding, Relationship and Physical Facility)
Lecture 2: Understanding Value Education
Lecture 3: Self-Exploration as the Process for Value Education
Lecture 4: Continuous Happiness and Prosperity – the Basic Human Aspirations
Lecture 5: Happiness and Prosperity – Current Scenario

UNIT II HARMONY IN THE HUMAN BEING 6 hours

Lecture 6: Understanding Human being as the Co-existence of the self and the body
Lecture 7: The body as an Instrument of the self
Lecture 8: Understanding Harmony in the self
Lecture 9: Harmony of the self with the body
Lecture 10: Programme to ensure self-regulation and Health

UNIT III HARMONY IN THE FAMILY AND SOCIETY 6 hours

Lecture 11: Harmony in the Family – the Basic Unit of Human Interaction
Lecture 12: 'Trust' – the Foundational Value in Relationship
Lecture 13: 'Respect' – as the Right Evaluation
Lecture 14: Other Feelings, Justice in Human-to-Human Relationship
Lecture 15: Understanding Harmony in the Society

UNIT IV HARMONY IN THE NATURE/EXISTENCE 6 hours

Lecture 16: Understanding Harmony in Nature
Lecture 17: Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature
Lecture 18: Realizing Existence as Co-existence at All Levels
Lecture 19: The Holistic Perception of Harmony in Existence

UNIT V IMPLICATIONS OF THE HOLISTIC UNDERSTANDING – A LOOK AT PROFESSIONAL ETHICS 6 hours

Lecture 20: Natural Acceptance of Human Values
Lecture 21: Definitiveness of (Ethical) Human Conduct
Lecture 22: Competence in Professional Ethics
Lecture 23: Holistic Technologies, Production Systems and Management Models-Typical Case Studies
Lecture 24: Strategies for Transition towards Value-based Life and Profession

Course Outcomes:

After completing this Unit, students will be able to
CO1: Understand the basic human aspiration and Natural Acceptance.
CO2: Aware of themselves and self-regulation.
CO3: Recognize human-human relationship (Justice) and identify human goals in society.
CO4: Appreciate the harmony in nature and existence.
CO5: Develop as socially and ecologically responsible engineers in handling problems with sustainable solutions (user-friendly and eco-friendly).

Text Book(s)

1. R R Gaur, R Asthana, G P Bagaria, *A Foundation Course in Human Values and Professional Ethics*, 3rd Revised Edition, UHV Publications, Sarva Shubha Nyas, Kanpur, 2023. ISBN: 978-81-957703-7-3 (Printed Copy) ISBN: 978-81-957703-6-6 (e-book)
2. R R Gaur, R Asthana, G P Bagaria, *Teachers' Manual for a Foundation Course in Human Values and Professional Ethics,* 3rd Revised Edition, UHV Publications, Sarva Shubha Nyas, Kanpur, 2023.

Reference Books

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book), Annie Leonard, Free Press, 2010.
4. The Story of My Experiments with Truth, Mohandas Karamchand Gandhi, Fingerprint! Publishing
5. Small is Beautiful - E. F Schumacher, Random House, 2011.
6. Slow is Beautiful - Cecile Andrews, New Society Publishers, 2006.
7. Economy of Permanence - J C Kumarappa
8. Vivekananda - Romain Rolland

Online Resources:

1. https://fdp-si.aicte-india.org/index.php
2. https://onlinecourses.swayam2.ac.in/aic22_ge23/preview
3. https://uhv.org.in/
4. https://www.youtube.com/@UniversalHumanValues/playlists
5. https://www.youtube.com/@mitsmadanapalle3058/playlists

Mode of Evaluation: Assignments, Mid Term Tests, End Semester Examination.

B. Tech I Year II Semester

25BAMATTC05 LINEAR ALGEBRA

L	T	P	C
3	0	0	3

Pre Requisites: 25BAMATTC02

Course Description:

The course provides a comprehensive foundation in linear algebra and its applications. The course explores the theory of matrices, vector spaces, linear transformations, and orthogonality, while also developing computational techniques essential for solving real-world problems in science and engineering.

Course Objectives:

1. To understand the types of matrices and techniques to solve the system of linear equations.
2. To interpret the significance of eigenvalues, eigenvectors and quadratic forms in Engineering problems.
3. To learn the fundamental concepts of vector spaces and bases.
4. To acquire skills in change of coordinatization and linear transformations.
5. To develop the techniques of orthogonality and projections.

UNIT I MATRICES AND SYSTEM OF LINEAR EQUATIONS 9 hours

Matrices- Orthogonal matrix, Idempotent matrix, Partition matrix, Permutation and Boolean Matrices, Reduced Row Echelon form, Rank, Inverses of matrices, System of linear equations, Gaussian Elimination, Gauss-Jordan method, Applications.

UNIT II EIGENVALUES AND EIGENVECTORS 9 hours

Eigenvalues and Eigenvectors, Properties, Diagonalization, Quadratic forms, Cayley-Hamilton theorem (without proof) - computing inverse and powers of a matrix.

UNIT III FINITE DIMENSIONAL VECTOR SPACE 9 hours

Vector spaces, subspaces, span, linear independence, basis and dimension, Coordinates with respect to ordered basis.

UNIT IV COORDINATIZATION AND LINEAR TRANSFORMATIONS 9 hours

Transition matrix for change of Coordinates, Introduction to Linear transformations, matrices of linear transformation, Kernel and Range, Dimension theorem, invertible linear transformations, applications to computer graphics.

UNIT V ORTHOGONALITY AND PROJECTIONS 9 hours

Real inner products, length, distance and angles, Gram-Schmidt orthogonalization, orthonormal bases, orthogonal projections, projection matrix, QR Factorization, Singular Value Decomposition.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Solve the systems of linear equations using matrix operations relevant to computational problems.
CO2: Determine eigenvalues and eigenvectors in real world computing tasks.
CO3: Utilize the concepts of vector space and basis in data modelling.
CO4: Construct linear transformations and compute transition matrices for coordinate transformations.
CO5: Apply orthogonal projections, orthonormal bases and SVD in image compression.

Text Books:

1. Elementary linear algebra, Stephen Andrilli & David Hecker, Academic Press (Elsevier), 4th Edition, 2012.

Reference Books:

1. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 42th Edition, 2012.

2. Linear Algebra with Applications, Gareth Williams, 9th Edition, Jones & Bartlett Learning, 2018.

3. Jin Ho Kwak and Sungpyo Hong, Linear Algebra, Birkhaüser, Second edition, 2004.

4. Ferrante Neri, Linear Algebra for Computational Sciences and Engineering, Springer, 2nd Edition, 2016.

5. Mark J. DeBonis, Introduction to Linear Algebra, CRC Press, 1st Edition, 2022.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech I Year II Semester

25BAPHYTC03 MODERN PHYSICS FOR QUANTUM TECHNOLOGIES

L	T	P	C
3	0	0	3

Pre-requisites: Plus two physics fundamental knowledge

Course Description:

This course offers an accessible and engaging introduction to the fundamental concepts of modern physics and their applications in quantum computing. specifically designed for computer science and engineering students. It explores key quantum concepts such as superposition, entanglement, and quantum gates, and explains how these ideas form the foundation of quantum information processing. Students will also explore the hardware side of quantum computing, including optics and photonics-based systems, linear optical components, semiconductor technologies, magnetism, and superconducting qubits used in modern quantum processors. The course combines theoretical understanding with practical insights into the tools, frameworks, and emerging technologies driving real-world quantum computation.

Course Objectives:

1. Understand and apply principles of interference, diffraction, and polarization in advanced optical systems for quantum technologies.
2. Interpret Schrödinger's equation to simple quantum systems and apply quantum tunneling phenomena.
3. Explain quantum superposition, entanglement, and quantum gates for computation.
4. Understand semiconductor physics, quantum Hall effect, and quantum dots in computing hardware.
5. Apply magnetism and superconductivity for superconducting qubits and quantum circuits.

UNIT I WAVE OPTICS 9 hours

Principles of interference, Young's double slit experiment, Newton's rings experiment, Michaelson interferometer, Quantum interference, Principles of Diffraction, Single Slit, double slit, diffraction grating (Qualitative). Polarization and Types, methods of Polarization, Nicol Prism, Beam splitters (Intensity & polarization), phase shifters (Half wave and Quarter wave).

UNIT II QUANTUM MECHANICS 9 hours

De Broglie's hypothesis, Uncertainty principle, Postulates of quantum mechanics - Wave function, Operators, Quantum states and measurement, Hilbert space, Schrodinger wave equations: Time dependent and independent, Applications of Schrodinger wave equation: free particle, particle in a box, and tunnelling.

UNIT III FOUNDATIONS OF QUANTUM COMPUTING 9 hours

Qubit, single and multi-qubit systems, Bloch sphere visualization, Quantum vs classical computers, Principles of superposition, Quantum Parallelism. Measurement of Quantum states and observation, Quantum Entanglement: introduction, EPR pairs and importance in quantum information, Unitary transformations, Elementary quantum gates.

UNIT IV FREE ELECTRON THEORY & SEMICONDUCTORS 9 hours

Quantum free electron theory, origin of energy bands, band structure, classification of materials, density of states, Fermi energy level.

Semiconductor types, intrinsic and extrinsic, Drift and Diffusion Current, Hall effect, 0-, 1-, 2-dimensional quantum materials, Quantum dots, Quantum Dots as Qubits, Quantum Hall Effect (QHE) and Sensors.

UNIT V MAGNETISM AND SUPERCONDUCTIVITY FOR QUANTUM TECHNOLOGIES **9 hours**

Magnetism: Atomic origin and classification of magnetic materials, hysteresis, magnons. Superconductors: Type I & II, BCS theory, Josephson junctions, superconducting qubits, Quantum circuits, SQUIDs, superconductors in quantum computers.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Apply optical interference, diffraction, and polarization principles in designing quantum optical systems.

CO2: Solve simple quantum systems using Schrödinger's equation and interpret tunnelling phenomena.

CO3: Represent and manipulate qubit states, superposition, and entanglement using basic quantum gates.

CO4: Understanding semiconductor properties and quantum effects for their role in quantum hardware.

CO5: Apply the magnetic and superconducting principles relevant to superconducting qubits and Quantum circuits.

Text Books:

1. Engineering Physics –Dr. M.N. Avadhanulu, Dr. P.G. Kshirsagar & TVS Arun Murthy, 11th editions 2018, S. Chand and Company.
2. Quantum Mechanics, Concepts and Applications (An Indian Adaptation) - **Nouredine Zettili,** 2ed: by Wiley.
3. Quantum Computing: A Beginner's Introduction, - Parag K. Lala, 1st Edition, 2019 McGraw-Hill Education.
4. Quantum Computing for Everyone – Bernhardt, 2020, The MIT Press
5. Engineering Physics - H.K. Malik & A.K. Singh, 2nd Edition, 2018, McGraw Hill Education (India) Private Limited.

Reference Books:

1. Concepts of Modern Physics, - Arthur Beiser, 7th Edition, 2017, TATA Mc Graw Hill
2. Engineering Physics –K. Thyagarajan, 2018, McGraw Hill Publisher
3. Physics Vol I & II, - Halliday/Resnick/Krane 5th Edition, 2007, John Wiley.
4. Quantum Computation and Quantum Information - Nielsen & Chuang, 10th Anniversary Edition, 2010, CAMBRIDGE UNIVERSITY PRESS
5. Solid State Electronic Devices- B.G. Streetman, Sanjay Kumar Banerjee, 7th ed. 2018, Pearson,
6. Superconductivity: Basics and Applications to Magnets - P. N. Barnes, 2015, Springer

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech I Year II Semester

25BACSETC03 DATA STRUCTURES AND ALGORITHMS

L	T	P	C
3	0	0	3

Pre-requisite C Programming for Problem Solving

Course Description:

This course introduces fundamental data structures, algorithm analysis, and performance evaluation. Students will gain hands-on experience in implementing linear and non-linear data structures, searching, sorting, hashing, and graph traversal techniques using C.

Course Objectives:

This course enables students to

1. Understand the fundamentals of data structures and algorithm complexity.
2. Learn different linear data structures (arrays, linked lists, stacks, queues) and their applications.
3. Study non-linear data structures such as trees and graphs with traversal techniques.
4. Analyze and implement searching, sorting, and hashing algorithms with performance evaluation.
5. Apply suitable data structures and algorithms to solve real-world problems effectively.

UNIT I INTRODUCTION TO DATA STRUCTURES 9 hours

Introduction to Data Structures, Abstract Data Types (ADTs), Complexity analysis: Asymptotic notation (O, Ω, Θ), Best, Worst and Average Case Analysis.

UNIT II LINKED LIST 9 hours

Singly, Doubly, and Circular Linked Lists – Operations and Applications.

UNIT III STACKS AND QUEUES 9 hours

Stack ADT: Definition, Operations, Applications.
Queue ADT: Definition, Operations, Circular queues, Double-ended queues (deque), Priority queues, Applications.

UNIT IV SEARCHING & SORTING 9 hours

Searching: Linear Search, Binary Search.
Sorting: Insertion sort, Selection sort, Quick sort, Bubble sort, Merge sort, Heap sort.
Best, Worst and Average Case Analysis of Searching & Sorting techniques.

UNIT V TREES & GRAPHS 9 hours

Trees: Binary Trees: Introduction, Traversals (Inorder, Preorder, Postorder), Binary Search Tree (BST) construction, insertion, deletion, AVL Tree.
Graphs: Basic terminologies, Breadth First Traversal, Depth First Traversal.

Course Outcomes:

After completing this Unit, students will be able to

CO1: Explain the concepts of data structures and evaluate algorithm complexities.

CO2: Implement and apply linear data structures for problem-solving.

CO3: Construct and manipulate non-linear data structures including trees and graphs.

CO4: Apply searching, sorting, and hashing algorithms with performance analysis.

CO5: Choose appropriate data structures and algorithms for solving practical computing problems.

Text Book(s)

1. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, Pearson Education. Ltd., Second Edition.

Reference Books

1. Data Structures, Algorithms and Applications in C by Sartaj Sahni, McGraw Hill, NY, Second Edition.
2. Data Structures and Algorithms, Alfred V. Aho, John E. Hopcroft, Jeffery D.Ulman. Pearson; 1st Edition, 2002.

Mode of Evaluation: Assignments, Mid Term Tests, Semester End Examination.

B. Tech I Year II Semester

25BACAMTC01 FUNDAMENTALS OF INTELLIGENT SYSTEMS

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Description:

The course introduces evolution of computing, characteristics, applications, challenges, and ethical aspects of intelligent systems. It also focuses on problem-solving techniques, planning methods, intelligent agents, and multi-agent systems.

Course Objectives:

1. To understand the evolution, components, characteristics, applications, challenges, and ethical aspects of intelligent systems.
2. To learn problem-solving methods using state-space representation and search strategies, including heuristic and randomized approaches.
3. To explore problem decomposition techniques and game-playing algorithms in intelligent systems.
4. To study various planning methods and their application in real-world problem solving.
5. To analyze intelligent agents, their architectures, multi-agent systems, and real-world applications of intelligent systems.

UNIT I INTRODUCTION TO INTELLIGENT SYSTEMS 9 hours

Evolution of computing: From conventional to intelligent systems, Defining AI and intelligent systems, Components of intelligent systems: Perception, reasoning, learning, action, Characteristics of intelligent behavior, Applications: Expert systems, NLP, robotics, vision, speech, AI vs Machine Learning vs Deep Learning, Challenges and limitations of intelligent systems, Ethics, bias, and societal impact of AI.

UNIT II PROBLEM SOLVING AND SEARCH STRATEGIES 9 hours

State Space Search: Simple Search, DFS, BFS, Heuristic Search: Heuristic functions, Best First Search, Hill Climbing, Local Maxima, Randomized Search: Escaping Local Maxima, Iterated Hill Climbing, Genetic Algorithms.

UNIT III PROBLEM DECOMPOSITION AND GAME PLAYING 9 hours

Problem Decomposition: SAINT, Dendral, Goal Trees, Rule Based Systems, Rule Based Expert systems, Game Playing: Board Games, Game Playing Algorithms, Limitations of Search, Other Games.

UNIT IV PLANNING METHODS 9 hours

Planning: The STRIPS Domain, Forward State Space Planning, Backwards State Space Planning, Goal Stack Planning, Advanced Planning Methods: Graph Plan, Planning as Constraint Satisfaction, Planning as Satisfiability, Heurisitic Search, Durative Actions, Planning in Rea1 World.

UNIT V INTELLIGENT AGENTS AND APPLICATIONS 9 hours

Definition of agents and agent environments, Types of agents: Simple reflex, model-based, goal-based, utility-based, Agent architectures: Reactive, deliberative, hybrid, Multi-Agent systems: Cooperation, competition, communication, Intelligent personal assistants (Siri, Alexa, Google Assistant), Recommender systems and personalization, Autonomous vehicles and robotics, Future of intelligent systems: Trends, opportunities, and challenges.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Analyze the evolution, components, characteristics, applications, and ethical issues of intelligent systems.

CO2: Apply state-space and heuristic search techniques to solve AI problems.

CO3: Analyze problem decomposition methods and game-playing algorithms for intelligent decision-making.

CO4: Apply planning methods to real-world problem-solving scenarios.

CO5: Analyze intelligent agent architectures and their applications in multi-agent and real-world systems.

Text Books:

1. Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson, 2021.
2. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education Pvt. Ltd.

Reference Books:

1. Adrian A. Hopgood, "Intelligent Systems for Engineers and Scientists: A Practical Guide to Artificial Intelligence", 4th Edition, CRC Press, 2021.
2. Charu C. Aggarwal, "Artificial Intelligence: A Textbook", Springer Cham, 2021.

Mode of Evaluation: Assignments, Mid Term Tests, Semester End Examination.

B. Tech I Year II Semester

25BAENGLC01 PROFESSIONAL COMMUNICATION LABORATORY

L	T	P	C
0	0	2	1

Course Description:

This course enhances proficiency in English communication with emphasis on fluency, accurate pronunciation, and confidence. It develops speaking, listening, reading, and writing skills through structured activities. Learners will also develop competence in composing formal correspondence, reports, presentations and designing effective posters.

Course Objectives:

This course aims to:

1. Enable students to confidently greet others, introduce themselves and others in professional and cross-cultural contexts, and describe their everyday routines, roles, environments, and familiar locations with clarity.
2. Build the skill to narrate past experiences, achievements or memorable events clearly and to articulate professional goals, plans and predictions for the future.
3. Enhance practical communication skills such as making enquiries, requesting, giving directions or offering clarifications, and drafting accurate reports in academic and workplace settings.
4. Strengthening reading and listening skills by engaging with stories to improve comprehension, vocabulary, and critical appreciation.
5. Prepare students to compose formal emails, reports, letters, and to make posters, and deliver presentations using effective English expressions, structures, and style.

Lab Activities

Activities shall include listening tasks and a revisit of grammar, vocabulary, pronunciation, and intonation (wherever required)

1. Introduction to sounds of English, stress and intonation for apt pronunciation
2. Greetings; Introducing oneself and others; conversations
3. Describing a person, place, object, etc.
4. Narrating a personal incident/events
5. Planning/predicting a future event/prospects
6. Skit on enquiring and requesting
7. Roleplays on giving formal/informal instructions and reporting what has been done so far in the lab / general scenario
8. Listening to select video/audio and discussing a favorite part of the story/ documentary in groups.
9. Reading comprehension and reading business emails
10. Writing email, letters/reports
11. Presentations and making posters (online and offline)

Course Outcomes:

Students will be able to:

CO1: Speak English fluently with a good pronunciation in an Indian accent, and confidently handle greetings,
introductions, and provide descriptions of their routines, locations, and surrounding objects.

CO2: Narrate stories, incidents, or personal experiences, as well as engage in planning and predicting future
possibilities using appropriate English structures.

CO3: Demonstrate effective English communication by asking questions, making polite requests, giving
instructions, and presenting reports or information clearly.

CO4: Apply listening and reading skills for comprehension, interpretation, and language development in both
academic and professional contexts.

CO5: Compose grammatically correct and well-structured formal emails, reports, presentations, and business
letters, and design posters with clear English expressions.

Text Books:

1. Communicative English – A Workbook by Shobha K.N. & Rayen J. Lourdes (Cambridge University Press, 2019)

Reference Books:

1. Communication Skills: A Workbook by Sanjay Kumar & Pushp Lata (Oxford University Press, 2019)
2. ELCS Lab Manual: A Workbook for CALL and ICS Lab Activities (Board of Editors, Orient Black Swan Pvt. Ltd., 2016)
3. English Language Skills: A Practical Approach by Veerendra Mishra et al. (Cambridge University Press, 2020)

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech I Year II Semester

25BAPHYLC01 PHYSICS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisites: Plus two physics knowledge

Course Description:

This course offers a comprehensive exploration of fundamental physics principles through a series of well-designed experiments, emphasizing both conceptual understanding and hands-on measurement techniques.

Course Objectives:

1. To experimentally analyze and quantify optical phenomena such as interference, diffraction, and polarization, using techniques like Newton's rings, single slit, and diffraction grating.
2. To determine and understand essential material and electronic properties, including energy gap in semiconductors and the mechanical strength of materials using specialized methods.
3. To measure and interpret magnetic and electromagnetic characteristics by studying B-H curves, magnetic fields, and resonance phenomena in electrical circuits.
4. To estimate fundamental physical constants such as Planck's constant and the charge-to-mass ratio of an electron using classical experimental setups.
5. To develop skills in conducting precise experimental investigations, analyzing data, and deriving key physical quantities relevant to modern physics and engineering applications.

List of Experiments:

1. Determination of radius of curvature of a given Plano-convex lens by Newton's rings.
2. Determination of wavelengths of different spectral lines in mercury spectrum using diffraction grating in normal incidence configuration.
3. To determine the slit width (d) using the diffraction pattern produced by a single slit.
4. To determine the Brewster's angle and refractive index of a glass.
5. Determination of Frequency of electrically maintained tuning fork by Melde's experiment.
6. Study the variation of B versus H by magnetizing the magnetic material (B-H curve).
7. Determination of wavelength of Laser light using diffraction grating.
8. Determination of energy gap of a semiconductor using p-n junction diode.
9. Magnetic field along the axis of a current carrying circular coil by Stewart Gee's Method.
10. Determination of young's modulus for the given material of wooden scale by non-uniform bending (or double cantilever) method.
11. Estimation of Planck's constant using photoelectric effect.
12. Determination of numerical aperture and acceptance angle of an optical fiber.
13. To study the frequency response of series LCR circuit-resonance frequency, band width and quality factor.
14. To find the charge to mass ratio of an electron using Thomson's method.

Note: Any TEN of the listed experiments are to be conducted. Out of which any TWO experiments may be conducted in virtual mode.

Course Outcomes:

CO1: Know the various phenomena of light practically and gain knowledge about various optical technique methods.
CO2: Verify the theoretical concepts of optics, magnetism, waves and oscillations by hands on experiment.
CO3: Apply the scientific process in the conducting of experiments and report the experimental findings.
CO4: Understand mechanical phenomena by instruments and apply them in real time applications.
CO5: Acquire and interpret experimental data to examine the physical laws.

Text / Reference Books:

1. A Textbook of Practical Physics - S. Balasubramanian, M.N. Srinivasan, S. Chand Publishers, 2017.
2. Engineering Physics Laboratory Manual, Jayaraman, 2013, Pearson Education.
3. A Course of Experiments with He-Ne Lasers, R.S. Sirohi, New Age International (P) limited, Publishers, 1985.

Web Resources: www.vlab.co.in
https://phet.colorado.edu/en/simulations/filter?subjects=physics&type=html,prototype

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech I Year II Semester

25BACSELC02 IT ESSENTIALS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: NIL

Course Description:

This course equips students with foundational computing skills, including computer hardware assembly, safe internet usage, professional document creation, cloud collaboration, and responsible use of AI tools for productivity and creativity.

Course Objectives:

1. Understand computer components, assembly, and OS installation.
2. Create documents, spreadsheets, and presentations using productivity tools.
3. Utilize cloud tools for storage, scheduling, and data collection.
4. Explore AI tools for prompts, creative tasks, and translation.
5. Build foundational computing skills with safe internet practices and responsible AI use.

List of Experiments:

1. **PC Hardware:**

Task 1: Hardware Identification

- Identify common computer peripherals (input, output, storage) and list Central Processing Unit (CPU) components with their functions.

2. **Internet & Email:**

Task 1: Networking Concepts and Safe Practices

- LAN, MAN, WAN, Wi-Fi, and the Internet.
- Web browsers, web navigation, search engines, web surfing best practices, netiquette, and cyber hygiene (Passwords, Updates, Phishing Awareness, Safe Downloads, Backups).
- Bookmark webpages and configure or showcase a pop-up blocker in a modern browser.

Task 2: Email Operations

- Create an email account, compose and send an email, and demonstrate unsubscribing from unwanted mailing lists using standard unsubscribe mechanisms or settings.

3. **Office Productivity Tools (MS-Office/Libre Office):**

Word Processor

Task 1: Event Certificate Creation

- Design a professional event certificate:
  - Font formatting (typeface, size, style), Drop Cap for emphasis, Text effects and character spacing adjustments, Borders and color schemes, Header and footer for branding/institutional details, Insert → Date and Time field for event date and Export final certificate to PDF.

Task 2: Newsletter Development

- Create a newsletter:
  - Automatic Table of Contents, Multi-column (newspaper) layouts, Images from local files and LibreOffice Clipart Gallery, Drawing tools and Fontwork for decorative headings, Textboxes for flexible content placement, Paragraph styles for consistent formatting, Built-in Mail Merge Wizard with spreadsheet data.

Spread Sheet

Task 1: Scheduler Creation

- Build a scheduler
    - Gridlines and cell formatting (dates, times, custom formats).
    - AutoFill for repeated patterns, Text formatting for emphasis.

Task 2: GPA Calculator & Visualization

- Create a GPA calculator:
    - Mathematical functions (sum, average, etc.)
    - Charts to visualize data, Hyperlinks for navigation between sheets or to online resources.

Task 3: LOOKUP/VLOOKUP

- Boolean and logical operators, Conditional formatting.

Presentation

Task 1: Interactive Presentation

- Develop an interactive presentation:
    - Insert Hyperlinks, Images, Audio, Video, and Clip Art.
    - Insert tables, and charts.
    - Custom animations, slide transitions, and action buttons.

Task 2: Student Project Presentation using CANVA .

4. **Cloud Storage and Calendar:**

Task 1: Data Management and Scheduling

- Store, organize, share, and retrieve files using a cloud storage service (e.g., Google Drive), including restoring items from Trash when applicable.
- Schedule activities using a calendar application (e.g., Google Calendar), including creating events, setting recurring schedules, inviting guests, attaching Drive files, and configuring notifications.

Task 2: Event Form Creation (Microsoft/Google/Any online platform)

5. **AI Tools & Ethical Usage:**

Task 1: Prompt Engineering and Response Evaluation

- Formulate clear, precise prompts or questions for an AI-based tool (e.g., ChatGPT, Gemini).

Task 2: AI-Assisted Creative Writing

- Provide an introductory sentence, paragraph, or descriptive scene to an AI text-generation tool.
- Allow the AI to continue the narrative, ensuring coherence in tone, style, and plot progression.
- Assess the creativity, originality, and narrative flow of the generated content.

Task 3: AI-Powered Language Translation and Plagiarism Check

- Input a sentence or passage in a source language into an AI translation tool.
- Translate it into a target language and review the results for grammar, fluency, and contextual meaning.
- Plagiarism Check for the AI-generated translation to human or reference translations.

Course Outcomes:

After completion of the course, Students will be able to,

CO1: Identify computer hardware, networking basics, and safe internet/email use.

CO2: Create documents, spreadsheets, and presentations using productivity tools.

CO3: Use advanced spreadsheet functions for data analysis and visualization.

CO4: Manage files, schedules, and forms with cloud applications.

CO5: Assess AI tool outputs for accuracy, creativity, and ethical use.

Reference Books:

1. Comdex Information Technology course tool kit, Vikas Gupta, WILEY Dream tech, 2003.
2. The Complete Computer upgrade and repair book, Cheryl A Schmidt, WILEY Dream tech,2013, 3rd edition.
3. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education, 2012, 2nd edition.
4. PC Hardware - A Handbook, Kate J. Chase, PHI (Microsoft).
5. IT Essentials PC Hardware and Software Companion Guide, David Anfins on and Ken Quamme– CISCO Press, Pearson Education, 3rd edition.
6. IT Essentials PC Hardware and Software Labs and Study Guide, Patrick Regan– CISCO Press, Pearson Education, 3rd edition.

Mode of Evaluation: Continuous Internal Evaluation, Model Test and Semester End Examination

B. Tech I Year II Semester

25BACSELC03 DATA STRUCTURES AND ALGORITHMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite C Programming for Problem Solving Laboratory

Course Description:

This course provides hands-on practice with fundamental data structures and algorithms using the C programming language. Students will design and implement linear and non-linear data structures, perform standard operations, and apply searching, sorting, and traversal techniques to solve real-world problems.

Course Objectives:

This course enables students to

1. Implement core data structures and algorithms using C language.
2. Develop competency in linear structures (arrays, linked lists, stacks, queues) and their operations.
3. Construct and manipulate non-linear structures (trees and graphs) with traversal techniques.
4. Analyze time and space complexity of implemented algorithms.
5. Apply suitable data structures to model and solve real-world problems.

List of Experiments:

1. (a) Given an integer array nums and an integer k ($1 \leq k \leq$ len(nums)), return the kth largest element in nums. The kth largest element is the element that would appear at index $k-1$ if the array were sorted in descending order.
   (b) Develop a program and compute the complexity analysis of reversing an one-dimensional array of length n.
2. Implement a menu-driven application that manages patients in a Hospital Management System. Each patient node should store relevant details (e.g., Patient ID, Name, Age, and Ailment/Department). The program must support the following operations and demonstrate them in sequence:
   a. Add at least five patient records to the queue (rear insertion).
   b. Delete the patient at the front of the queue (beginning).
   c. Delete a patient from the middle of the queue (by Patient ID or position), ensuring list integrity is maintained.
   d. Search for a specific patient by a unique identifier (e.g., Patient ID) and display the patient's details if found.

Display the final state of the queue after performing the above operations, showing all remaining patients in order.

3. Build a menu-driven application that maintains a playlist where each node represents a song. The program should support the following operations and demonstrate:
   a. Add at least five songs to the playlist (tail insertion recommended).
   b. Delete one song from the beginning (head deletion).
   c. Delete one song from the middle (by Song ID or position), ensuring all links are updated correctly.
   d. Display the playlist in forward direction (head to tail).

e. Display the playlist in backward direction (tail to head).

Search for a particular song by its unique Song ID and display its details if found.

4. Implement a menu-driven application that manages page visits. Each entry represents a web page. Demonstrate the following sequence within the program:
   a. Start with an empty history stack.
   b. Visit 5 pages by pushing their identifiers/URLs onto the stack.
   c. Press "Back" twice by performing two pop operations.
5. Develop a Ticket Counter System for a cinema hall where customers are served in the same order they arrive. Implement the following steps:
   a. Add at least 5 customers to the queue.
   b. Serve (dequeue) two customers.
   c. Peek the next customer to be served.
   d. Display the full waiting list.
6. Develop a menu-driven application that manages a directory tree. Each node stores:
   a. int id (unique identifier)
   b. int isFolder (1=folder, 0=file)
   c. char name[...] (optional but recommended)
   d. struct Node* left, *right

Implement the following tasks:

   i. Insert at least seven nodes (file/folder IDs) into the tree. Use a consistent insertion policy (by id).
   ii. print all the traversal methods (Preorder, Inorder, Postorder).
   iii. Search for a specific file/folder by ID and display whether it exists along with its type (file/folder) and name if available.
   iv. Delete a folder (by ID) from the tree. Ensure standard BST deletion behavior (0, 1, or 2 children) and maintain tree integrity.
   v. Show the final directory structure using Inorder traversal with clear labeling.

7. Develop a program that simulates incoming player scores and maintains a descending (highest-to-lowest) leaderboard using insertion sort. The program should:
   a. Implement a function insertDescending(int arr[], int n, int newScore) that inserts newScore into its correct position using insertion sort logic.
   b. Print the leaderboard after each insertion with ranks.

8. Develop an application that accepts a set of athlete completion times recorded in random order and arranges them in ascending order (smallest time = fastest).
   a. Use an array of integers or floats to store completion times.
   b. Implement Selection Sort to order times in ascending order.
   c. Print the list before and after sorting.
   d. Display the top performers.
9. Design and implement a program that models a city's bus transit network as an unweighted graph and, given a starting stop, lists all reachable stops by minimum number of bus rides and computes a shortest path (in hops) to a specified destination.
10. Design and implement a program that determines whether a given word exists in a 2D grid of letters by performing Depth-First Traversal (DFT).

Course Outcomes:

After completion of the course, students will be able to

CO1: Implement the core array operations and analyze the time-space complexities.

CO2: Apply linear and non-linear data structures algorithms for the real-world problem solving applications.

CO3: Construct and manipulate tree-based data structures—binary trees and binary search trees—and perform traversal, insertion, and deletion operations.

CO4: Implement and compare sorting algorithms with respect to time and space complexity.

CO5: Apply graph traversal algorithms to explore and process graph structures.

Text Books:

1. Data Structures, Algorithms and Applications in C by Sartaj Sahni, McGraw Hill, NY, Second Edition.

References Books:

1. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, Pearson Education. Ltd., Second Edition.
2. Data Structures and Algorithms, Alfred V. Aho, John E. Hopcroft, Jeffery D.Ulman. Pearson; 1st Edition, 2002.

Mode of Evaluation: Continuous Internal Evaluation and Semester End Examination.

II Year I Semester

B. Tech II Year I Semester

25BAMATTC09 PROBABILITY AND STATISTICS FOR COMPUTER SCIENCE

L	T	P	C
3	0	0	3

Pre-requisite 25BBMATTC01

Course Description:

This course provides a comprehensive foundation in probability and statistics with applications in computer science and data analysis. It begins with the fundamentals of probability and random variables, followed by the study of standard probability distributions and their applications. The course then explores joint distributions and dependency analysis for multivariate data. It further emphasizes statistical data analysis techniques, including descriptive measures, correlation, and regression for predictive modelling. Finally, the course covers statistical inference methods such as estimation, confidence intervals, and hypothesis testing for data-driven decision making.

Course Objectives:

The objectives of this course are:

1. To develop a strong foundation in probability concepts and random variable modelling.
2. To apply standard probability distributions for modelling uncertainty in real-world problems.
3. To analyse multivariate data using joint distributions and dependency measures.
4. To perform statistical data analysis using descriptive, correlation, and regression techniques for predictive modelling.
5. To apply statistical inference methods, including estimation and hypothesis testing, for data-driven decision making.

UNIT I PROBABILITY AND RANDOM VARIABLES 9 hours

Introduction to probability, axioms of probability, addition theorems, conditional probability, multiplication theorem, independence of events, Bayes' theorem.

Random variables (discrete and continuous), probability density functions, cumulative distribution function (CDF), mathematical expectation, properties of expectation, moments and moment generating functions.

UNIT II PROBABILITY DISTRIBUTIONS AND APPLICATIONS 9 hours

Discrete probability distributions: Binomial, Poisson, and Geometric distributions — properties and applications.

Continuous probability distributions: Uniform, Exponential, Gamma, and Normal distributions — properties and applications.

UNIT III JOINT DISTRIBUTIONS AND DEPENDENCY ANALYSIS 9 hours

Joint probability distributions (discrete and continuous), marginal distributions, conditional distributions. Independence of random variables, expectation of joint distributions, covariance and correlation.

Transformation of random variables (basic methods for one and two variables).

UNIT IV STATISTICAL DATA ANALYSIS AND PREDICTIVE MODELLING 9 hours

Descriptive analysis: Moments (raw and central moments), measures derived from moments Distribution analysis: skewness and kurtosis to understand the shape and characteristics of data - Relationship analysis: Karl Pearson's coefficient of correlation, rank correlation - Regression analysis: principle of least squares, simple linear regression model, regression coefficients and their properties

UNIT V HYPOTHESIS TESTING AND STATISTICAL DECISION MAKING 9 hours

Population and sampling, formulation of null and alternative hypotheses, level of significance, Type I and Type II errors, power of a test.

Large sample tests: test for single mean, single proportion, difference of means, difference of proportions. Confidence intervals for parameters (one-sample and two-sample cases).

Small sample tests: t-test for single mean, difference of means, and F-test for ratio of variances.

Course Outcomes:

At the end of the course, students will be able to:

CO1: Apply probability concepts and random variables to model uncertain events. (L3 – Apply)

CO2: Analyze and use appropriate probability distributions for real-world applications. (L4 – Analyze)

CO3: Evaluate relationships among variables using joint distributions, covariance, and correlation. (L4 – Analyze)

CO4: Perform data analysis using descriptive statistics and regression techniques for prediction. (L3 – Apply)

CO5: Apply statistical inference methods, including confidence intervals and hypothesis testing for decision making. (L3 – Apply)

Text Books

1. Milton. J. S. and Arnold. J.C., "Introduction to Probability and Statistics", Tata McGraw Hill, 4th Edition, 2007.
2. Dr.B.S.Grewal, " Higher Engineering Mathematics", Khanna Publications, 42nd Edition.

Reference Books

1. Spiegel. M.R., Schiller. J. and Srinivasan. R.A., "Schaum's Outline of Theory and Problems of Probability and Statistics", Tata McGraw Hill Edition, 2004.
2. Devore. J.L., "Probability and Statistics for Engineering and the Sciences", Cengage Learning, New Delhi, 8th Edition, 2012.
3. Dean G. Duffy., "Advanced Engineering Mathematics with MATLAB", CRC Press, Third Edition 2013.

E Books

1. http://nptel.ac.in/courses/IIT-MADRAS/Principles_of_Communication1/Pdfs/1_5.pdf
2. https://www.khanacademy.org

Mode of Evaluation: Assignments, Mid Term Tests, End Semester Examination.

B. Tech II Year I Semester

25BACOMTC01 COMPUTER ORGANIZATION AND ARCHITECTURE

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Description:

This course aims at introducing the concepts of computer architecture and organization. It involves design aspects, and deals with the current trends in computer architecture. It also aims to improve system performance by effective utilization of system resources such as memory and I/O subsystems.

Course Objectives:

1. To make students understand the basic structure and operation of digital computer.
2. To understand the hardware-software interface.
3. To familiarize the students with arithmetic and logic unit and implementation of fixed point and floating-point arithmetic operations.
4. To expose the students to the concept of pipelining.
5. To familiarize the students with hierarchical memory system including cache memories and virtual memory.
6. To expose the students with different ways of communicating with I/O devices and standard I/O interfaces

UNIT I OVERVIEW & INSTRUCTIONS 9 hours

Eight ideas – Components of a computer system – Technology – Performance – Power wall – Uniprocessors to multiprocessors; Instructions – operations and operands – representing instructions – Logical operations – control operations – Addressing modes.

UNIT II ARITHMETIC OPERATIONS 9 hours

Signed/Unsigned integer representation- ALU - Addition and subtraction – Multiplication – Sequential multiplication- Booths Algorithm- Modified Booths Algorithm- Division- restoring and non-restoring division – Floating point representation- floating point arithmetic – floating point addition/subtraction- floating point multiplication/division.

UNIT III PROCESSOR AND CONTROL UNIT 9 hours

Basic MIPS implementation – Building datapath – Pipelining – Pipelined datapath and control – Handling Data hazards & Control hazards: Dynamic Branch Prediction – Exceptions.

UNIT IV PARALLELISM 9 hours

Instruction-level-parallelism: Static and dynamic multiple issue processors – Parallel processing challenges – Flynn's classification – Hardware multithreading – Multicore processors.

UNIT V MEMORY AND I/O SYSTEMS 9 hours

Memory hierarchy - Memory technologies – Cache basics – Cache Mapping Techniques - Measuring and improving cache performance - Virtual memory, TLBs - Input/output system, programmed I/O, DMA and interrupts, I/O processors.

Course Outcomes:

Upon successful completion of the course, students will be able to
CO1: Understanding of the fundamental of computing systems.
CO2: Analyse the basic arithmetic and logical operations.
CO3: Analyse pipelining and hazards.
CO4: Understand parallel processing architectures.
CO5: Evaluate performance of memory and I/O systems.

Text Books:

1. David A. Patterson and John L. Hennessey, "Computer organization and design", Morgan Kauffman / Elsevier, Fifth edition, 2014.
2. V.Carl Hamacher, Zvonko G. Varanesic and Safat G. Zaky, "Computer Organisation", VI th edition, Mc Graw-Hill Inc, 2012

Reference Books:

1. William Stallings "Computer Organization and Architecture" , Seventh Edition , Pearson Education, 2006.
2. Vincent P. Heuring, Harry F. Jordan, "Computer System Architecture", Second Edition, Pearson Education, 2005.
3. Govindarajalu, "Computer Architecture and Organization, Design Principles and Applications", first edition, Tata McGraw Hill, New Delhi, 2005.
4. John P. Hayes, "Computer Architecture and Organization", Third Edition, Tata Mc Graw Hill, 1998.

Mode of Evaluation: Assignments, Mid Term Tests, End Semester Examination.

B. Tech II Year I Semester

25BACAMTC02 PYTHON PROGRAMMING

L	T	P	C
3	0	0	3

Prerequisites: NIL

Course Description:

This course introduces the fundamentals of Python programming, including data types, operators, control structures, data structures, functions, strings, modules, and file handling. It also covers exception handling, object-oriented programming, packages, stacks, and queues to develop efficient and modular Python applications.

Course Objectives:

1. Understand basic Python programming constructs.
2. Apply conditional statements and loops in Python to develop efficient problem-solving programs.
3. Understand Python functions, data structures, and variables to build efficient programs.
4. Learn string handling, modules, and file operations for efficient data processing in Python.
5. Apply exception handling, OOP concepts, and packages to build Python applications.

UNIT I INTRODUCTION TO PYTHON PROGRAMMING 9 hours

Introduction: Algorithms, building blocks of algorithms (flow chart), History of Python, Features of Python Programming, Variables, Assignment, Keywords, Input – Output, Comments, Running Python Scripts, Indentation, Python Data Types.

UNIT II OPERATORS AND EXPRESSIONS 9 hours

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 – else, ladder if, nested if – else, for, while, break, continue, pass.

UNIT III DATA STRUCTURES AND FUNCTIONS 9 hours

Lists, Tuples, Sets, Dictionaries, Functions - Defining Functions, Calling Functions, Passing Arguments, Variable in python – Global and Local Variables.

UNIT IV STRINGS, DIRECTORIES AND FILES 9 hours

String Handling – Modules: Creating Modules (datetime, time, OS, calendar, math module), import statement, from import statement, Name spacing, Directories, Files – File Path – Opening and Closing Files – Reading and Writing Files

UNIT V EXCEPTION HANDLING, INTRODUCTION TO OBJECT-ORIENTED CONCEPTS AND PACKAGES 9 hours

Exception: Errors and Exceptions, Exception Handling, Multiple Exceptions, Object Oriented Concepts: Class, Object, Inheritance, Polymorphism, Encapsulation and Abstraction, Python packages: User-defined Packages, Package Creation, Stacks and Queues: Introduction, Operations on Stack and Queue, Applications of Stack and Queue.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Understand the fundamentals of Python programming, algorithms, flowcharts, and data types.
CO2: Apply operators, expressions, and control structures to develop Python programs.
CO3: Analyze Python data structures and functions for problem solving and modular programming.
CO4: Implement string handling, modules, and file operations in Python applications.
CO5: Develop Python applications using exception handling, object-oriented concepts, and packages.

Text Books:

1. "Programming and Problem Solving with Python", Ashok Namdev Kamthane, Amit Ashok Kamthane, 2nd Edition, McGraw Hill, 2023
2. "Core Python Programming", Dr. R. Nageswara Rao, 3rd Edition, Dreamtech Press, 2021
3. "Python Programming: Using Problem Solving Approach", Reema Thareja, 2nd Edition, Oxford University Press, 2019

Reference Books:

1. "Let Us Python", Yashavant Kanetkar, Aditya Kanetkar, 2nd Edition, BPB Publications, 2019.
2. "Python: The Complete Reference", Martin C. Brown, 1st Edition, McGraw Hill Education, 2018

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech II Year I Semester

25BACAMTC03 ALGORITHM DESIGN AND COMPLEXITY ANALYSIS

L	T	P	C
3	0	0	3

Prerequisites: 25BACSETC03

Course Description:

This course introduces fundamental techniques for designing efficient algorithms to solve computational problems. It covers algorithmic paradigms such as divide-and-conquer, greedy methods, dynamic programming, and backtracking. The course emphasizes the analysis of time and space complexity using asymptotic notations and performance evaluation techniques. Students will develop problem-solving skills and learn to design, analyze, and compare algorithms for a variety of applications.

Course Objectives:

1. Analyze the asymptotic performance of algorithms and solve complex optimization problems efficiently.
2. Demonstrate various strategies in algorithm design using divide and conquer and greedy methods.
3. Understand dynamic programming techniques to optimize the solution of the given problems.
4. Demonstrate various backtracking techniques to solve complex problems.
5. Analyze the complexity measures to solve complex problems.

UNIT I INTRODUCTION AND PERFORMANCE ANALYSIS 9 hours

Introduction: Performance analysis, Asymptotic Notations & Basic Efficiency Classes, Asymptotic analysis of complexity bounds: best, average, and worst-case behavior, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

UNIT II DIVIDE AND CONQUER AND GREEDY METHOD 9 hours

Divide and Conquer: The General Method, Quick Sort, Merge Sort, Strassen's matrix multiplication. Greedy Method: General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees.

UNIT III DYNAMIC PROGRAMMING 9 hours

Dynamic Programming: General Method, All pairs shortest paths, Single Source Shortest Paths– General Weights (Bellman-Ford Algorithm), 0/1 Knapsack, Travelling Salesperson problem.

UNIT IV BACKTRACKING AND BRANCH AND BOUND 9 hours

Backtracking: General Method, 8-Queens Problem, Sum of Subsets problem, Graph Coloring.
Branch and Bound: The General Method, 0/1 Knapsack Problem, Travelling Salesperson problem.

UNIT V NP-HARD AND NP-COMPLETE PROBLEMS 9 hours

NP-Hard and NP-Complete Problems: Basic Concepts, Cook's theorem. NP-Hard Graph Problems: Clique Decision Problem (CDP), Chromatic Number Decision Problem (CNDP). NP-Hard Scheduling Problems.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Evaluate algorithms, expressing their efficiency using advanced notation
CO2: Develop effective algorithms by employing Greedy methods.
CO3: Develop optimized solutions using Dynamic Programming techniques.
CO4: Employ backtracking methods to optimize solutions.
CO5: Analyze the complexity measures of complex problems.

Text Books:

1. "Introduction to Algorithms". Thomas H Cormen, Charles E Lieserson, Ronald L Rivest, Clifford Stein, MIT Press, Fourth Edition, 2022.
2. "Fundamentals of Computer Algorithms", Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Orient Black Swan, 2nd Edition, 2008.

Reference Books:

1. "Algorithm Design: Foundations, Analysis, and Internet Examples", Michael T. Goodrich, Roberto Tamassia, John Wiley & Sons, Inc, 2nd Edition, 2009.
2. "Algorithm Design", Eva Tardos, Jon Kleinberg, Pearson Edition, 1st Edition, 2014

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech II Year I Semester

25BACAMTC04 OPERATING SYSTEM CONCEPTS

L	T	P	C
3	0	0	3

Prerequisites: NIL

Course Description:

This course introduces the fundamental concepts of operating systems, including operating system structures, shell scripting, processes, threads, CPU scheduling, process synchronization, and deadlock handling. It also covers memory management, virtual memory, page replacement, file system implementation, and disk scheduling, enabling students to apply these concepts for efficient system resource management.

Course Objectives:

1. Understand operating system structure and shell scripts
2. Apply concepts of threads and CPU scheduling algorithms
3. Analyze process synchronization techniques and deadlock handling
4. Apply memory allocation and page replacement techniques
5. Apply file system implementation methods and disk scheduling algorithms

UNIT I OPERATING SYSTEM STRUCTURE AND SHELL SCRIPTS 9 hours

Operating System structure: Operating System Services, User and Operating System interface, System calls, System Services; Basic Script Concepts, Expressions, Decisions: Making Selections, Repetition, Special Parameters and Variables, Changing Positional Parameters, Argument Validation, Debugging Scripts.

UNIT II PROCESS CONCEPTS 9 hours

Processes: Definition, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads. Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling.

UNIT III PROCESS SYNCHRONIZATION AND DEADLOCK HANDLING 9 hours

Process Synchronization: The Critical section problem, Race Conditions, Mutual Exclusion, The Producer Consumer Problem, Semaphores, Peterson's solution, Synchronization Tools, Semaphores, Monitors, Inter Process Communication (IPC), Message Passing, Classical problems of synchronization, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, avoidance, detection and recovery from deadlock.

UNIT IV MEMORY MANAGEMENT 9 hours

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, Fixed and variable partition, Internal and External fragmentation and Compaction; Paging: Principle of operation, Page allocation, Virtual Memory: Basics of Virtual Memory, Hardware and
control structures, Locality of reference, Page fault, Working Set, Dirty page/Dirty bit, Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not Recently Used (NRU) and Least Recently used (LRU).

UNIT V FILE SYSTEM AND DISK SCHEDULING 9 hours

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, File Allocation methods (contiguous, linked, indexed), Free-space management (Bit vector, linked list, grouping), Directory implementation (linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling – Virtualization and Containers.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Understand the structure of operating systems and shell scripting concepts.
CO2: Apply concepts of threads and CPU scheduling techniques for process execution.
CO3: Analyze process synchronization methods and deadlock management techniques.
CO4: Apply memory allocation methods and page replacement strategies.
CO5: Apply file system implementation techniques and disk scheduling methods.

Text Books:

1. "Operating System Concepts", Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, 10th Edition, John Wiley and Sons Inc., 2018
2. "Modern Operating Systems", Andrew S Tanenbaum, Pearson, 5th Edition, Pearson India, 2024

Reference Books:

1. "Operating systems - A concept based Approach", D.M Dhamdhere, 2nd Edition, Tata McGraw-Hill, 2002.
2. "UNIX and Shell Programming", Behrouz A. Forouzan, Richard F. Gilberg, 1st Edition, Cengage India Private Limited, 2003.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech II Year I Semester

25BACAMLC01 PYTHON PROGRAMMING LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Nil

Course Description:

This course provides hands-on experience in Python programming through problem-solving exercises involving control structures, functions, data structures, strings, file handling, and object-oriented programming. It also introduces statistical applications, stacks and queues, and Turtle graphics to develop practical Python programming skills.

Course Objectives:

1. To understand the fundamentals of Python programming and problem-solving techniques.
2. To develop Python programs using decision-making, looping, and functions.
3. To apply data structures, file handling, and string operations in Python applications.
4. To implement object-oriented programming concepts and exception handling techniques.
5. To develop Python applications using libraries, data structures, and Turtle graphics.

List of Programs:

1. a. In Wipro, a software developer noticed that two employee salary values were stored in the wrong variables. Develop a Python program to swap the two numbers:
   - Using a temporary variable
   - Without using a temporary variable

   b. A student from Jawaharlal Nehru Technological University wants to calculate the total number of assignments completed over n days. Write a Python program to find the sum of the first n natural numbers.

   c. Alexa travelled 150 kilometers by train while visiting Bengaluru. Write a Python program to convert the distance travelled from kilometers to miles.
2. a. A school management system stores the ages of Ram, Sam, and Khan. Write a Python program to determine the eldest and youngest among the three students.

   b. A digital security system checks whether an entered numeric passcode reads the same forward and backward. Write a Python program to check whether the given number is a palindrome or not.

   c. A mathematics student needs to calculate factorial values while solving permutation problems.

   d. A banking application needs to reverse account verification numbers and calculate the sum of digits for validation purposes. Write a Python program to find the reverse of a number and the sum of its digits.
3. a. A cryptography engineer at Google requires prime numbers for secure encryption algorithms. Write a Python script to generate prime numbers up to n.

   b. A primary school teacher wants to generate multiplication tables automatically for students. Write a Python program to generate the multiplication table of a given number.

   c. A mathematics learning application wants to identify special numbers for practice exercises. Write a Python program to determine whether a given number is an Armstrong number or not.
4. a. An online shopping application stores product IDs in a list. Write a Python script to:
   - Create a list

- Access elements
- Slice the list
- Add or modify elements
- Delete elements from the list

b. A university database stores student roll numbers in tuples because they should not be modified accidentally. Write a Python program to read values from a tuple and perform various operations.
c. A library management system stores book details using dictionaries. Write a Python script to perform dictionary operations such as insert, delete, and display.

5. a. A receptionist in a company office needs a computerized telephone directory to manage employee contact numbers. Write a Python program using dictionary and list to perform:
   - Add entry
   - Search entry
   - Delete entry
   - Update entry
   - View entries
   - Exit

   b. A data scientist at IBM is analyzing sales data and needs statistical measures to understand data distribution. Write a Python program to compute:
   - Mean
   - Variance
   - Standard Deviation
   - Skewness
   - Kurtosis

   c. A research analyst working on probability models needs to calculate Binomial and Poisson distributions for experimental data. Write a Python program to calculate Binomial and Poisson probability distributions for given input parameters.
6. a. A customer support executive at Amazon needs to process customer names and messages using string operations. Write a Python program to perform various operations on strings using string library functions such as:
   - Uppercase conversion
   - Lowercase conversion
   - String length
   - Replace operation
   - Splitting strings
   - Checking substrings
7. a. A mathematics teacher wants to demonstrate the Fibonacci sequence to students in a programming lab. Define a Python function that generates the Fibonacci series up to n numbers.
   b. A sports coordinator entered the scores of three players and wants to identify the highest scorer automatically. Define a function max_of_three() that takes three numbers as arguments and returns the largest among them.
   c. A data analyst working at Accenture needs to filter numbers based on specific conditions for reporting purposes. Write a Python program using functions to display all numbers between 1000 and 2000 that are divisible by 7 but not multiples of 5.

8. a. An office administrator wants to read and display employee records stored in a text file. Write a Python script to display the contents of a file.
   b. A backup management system needs to copy data from one file to another for safety purposes. Write a Python script to copy file contents from one file to another.
   c. A system administrator at Microsoft manages project folders and files in a computer system. Write Python commands to perform the following directory operations:
   - List directories and files
   - Create a new directory
   - Rename a directory or file
   - Remove a directory or file
9. a. An architect working at Larsen & Toubro needs software to calculate the area of rectangular rooms while designing building layouts. Define a class named Rectangle that can be constructed using length and width. The class should contain a method to compute the area of the rectangle.
   b. A civil engineering student is developing a geometry calculator to compute measurements of circular objects such as wheels and pipes. Define a class named Circle that is constructed using radius. Create derived classes Area and Circumference that use methods calArea() and calCirc() respectively to calculate the area and circumference of the circle.
10. a. A web browser maintains recently visited pages using stack operations. Develop a Python script to implement Stack ADT and perform the following operations:
    - Push
    - Pop
    - Peek
    - Display

    b. A railway reservation system processes passengers in the order they arrive at the ticket counter. Develop a Python script to implement Queue ADT and perform the following operations:
    - Enqueue
    - Dequeue
    - Display

    c. A graphics designer at Adobe is creating simple geometric designs for educational software. Using Turtle graphics in Python, write functions to draw the following shapes:
    - Triangle
    - Rectangle
    - Polygon
    - Circle
    - Square

COURSE OUTCOMES:

At the end of this course, students will be able to

CO1: Develop Python programs using basic syntax, operators, and control structures.
CO2: Apply functions, strings, lists, tuples, sets, and dictionaries in problem solving.
CO3: Implement file handling and directory operations using Python modules and libraries.
CO4: Develop Python applications using OOP concepts, exception handling, stacks, and queues.
CO5: Design simple graphical and statistical applications using Turtle graphics and Python libraries.

Text Books:

1. "Programming and Problem Solving with Python", Ashok Namdev Kamthane, Amit Ashok Kamthane, 2nd Edition, McGraw Hill, 2023.
2. "Core Python Programming", Dr. R. Nageswara Rao, 3rd Edition, Dreamtech Press, 2021.
3. "Python Programming: Using Problem Solving Approach", Reema Thareja, 2nd Edition, Oxford University Press, 2019.

Reference Books:

1. "Python Crash Course", Eric Matthes, 3rd Edition, No Starch Press, 2023.
2. "Automate the Boring Stuff with Python", Al Sweigart, 2nd Edition, No Starch Press, 2019.

Mode of Evaluation: Continuous Internal Evaluation of the Lab Experiments, Record, Viva-voce, and External Lab Examination

B. Tech II Year I Semester

25BACAMLC02 ALGORITHM DESIGN AND COMPLEXITY ANALYSIS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: 25BACSETC03

Course Description:

This laboratory course provides hands-on experience in designing, implementing, and analyzing algorithms for computational problems. Students will develop programs for algorithmic paradigms such as divide-and-conquer, greedy methods, dynamic programming, and backtracking. The course emphasizes experimental evaluation of algorithm performance through time and space complexity analysis.

Course Objectives:

1. To understand the basic concepts of algorithm design and analysis.
2. To apply sorting and searching techniques to solve computational problems.
3. To develop programs using graph traversal and shortest path algorithms.
4. To analyze the efficiency of algorithms using time complexity concepts.
5. To implement problem-solving techniques such as divide-and-conquer and backtracking.

List of Programs:

1. An e-commerce company maintains millions of product records. During festive sales, customers search for products continuously, and the system must return results efficiently. The development team wants to compare the performance of different searching techniques under varying input sizes and determine the best approach for large-scale systems. Students are hired as algorithm analysts to experimentally evaluate algorithm efficiency and identify the most scalable approach. Analyze best-case, average-case, and worst-case behavior.
2. A logistics company receives millions of parcels daily from different regions. To ensure timely delivery, parcels must be sorted rapidly according to destination codes. The company wants to compare different divide-and-conquer sorting techniques for scalability and efficiency. Compare performance for Random inputs, Sorted inputs and Reverse sorted inputs.
3. A cloud service provider receives multiple client jobs with deadlines and associated profits. Due to limited computing resources, not all jobs can be executed. The company wants to maximize revenue while meeting deadlines. Implement Job Sequencing with Deadlines. Schedule jobs to maximize overall profit. Implement Fractional Knapsack using Greedy Method.
4. A cloud service provider receives multiple client jobs with deadlines and associated profits. Due to limited computing resources, not all jobs can be executed. The company wants to maximize revenue while meeting deadlines. Implement Job Sequencing with Deadlines. Schedule jobs to maximize overall profit. Implement 0/1 Knapsack using Dynamic Programming.
5. During flood relief operations, emergency vehicles have limited carrying capacity. Relief materials such as medicines, food packets, and water bottles have different importance values and weights. Authorities need an optimized loading strategy. Implement suitable Greedy technique.
6. A robotics company is developing an autonomous rescue robot capable of navigating through complex maze-like environments during earthquake and disaster rescue operations. The robot must identify a valid path from the entry point to trapped victims while avoiding blocked or unsafe routes. Since multiple paths may exist and several routes can lead to dead ends, the robot requires a systematic trial-and-error mechanism to explore paths and backtrack whenever it encounters obstacles. The engineering team decides to use the Backtracking technique for pathfinding and navigation. Design and implement the navigation algorithm for the rescue robot.
7. There have been a number of fire outbreak cases recorded in the Florida area that has brought about loss of lives to inhabitants and loss of properties. Some routes within the district can be reconstructed into shortcut routes, so that fireman can traverse through the district in order to prevent fire incidents.

The objective is to find the minimum distance and shortest path from the fire station to all the residential layout in Florida area. Implement by applying Floyd's method to find the solution for the given scenario.

8. Bangalore Water supply Board responsibility is to distribute water evenly among all the areas in Bangalore city. A new layout has been developed by Max worth real estate developers. BWSB should connect the water lines to the new layout with minimum cost. The objective is to connect the water pipes so that it reaches all the houses in new layout with minimum cost. Implement by applying Kruskal's method to find the minimum spanning tree for the given scenario.
9. DigiMap services is a module in G-Maps which is used to find the distance from one place to another or from your location to the nearest desired location. This requires the Shortest Path Algorithm, as there are various routes/paths connecting them but it has to show the minimum distance. Represent a city/place with a vertex and the route between two cities/places as an edge, then by using Dijkstra's algorithm, find the shortest routes between any two cities/places or from one city/place to another city/place.
10. Consider the n-queens puzzle in which the goal is to place N queens on an N×N chessboard such that no two queens attack each other. A queen can attack horizontally, vertically, or diagonally. Given an integer N, return all distinct solutions to the N-queens puzzle. Note: Use Backtracking technique.

COURSE OUTCOMES:

At the end of this course, students will be able to

CO1: Implement fundamental sorting and searching algorithms using C programming.
CO2: Apply graph algorithms to solve real-world network and pathfinding problems.
CO3: Analyze the performance of algorithms in best, average, and worst cases.
CO4: Design efficient solutions using divide-and-conquer and greedy strategies.
CO5: Develop programs for complex problems using backtracking and pattern matching techniques.

Text Books:

1. "Introduction to Algorithms". Thomas H Cormen, Charles E Lieserson, Ronald L Rivest, Clifford Stein, , MIT Press, Fourth Edition, 2022.
2. "Fundamentals of Computer Algorithms", Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Orient Black Swan, 2nd Edition, 2008.

Reference Books:

1. "Algorithm Design: Foundations, Analysis, and Internet Examples", Michael T. Goodrich, Roberto Tamassia, John Wiley & Sons, Inc, 2nd Edition, 2009.
2. "Algorithm Design", Eva Tardos, Jon Kleinberg, Pearson Edition, 1st Edition, 2014

Mode of Evaluation: Continuous Internal Evaluation of the Lab Experiments, Record, Viva-voce, and External Lab Examination

B. Tech II Year I Semester

25BACAMLC03 OPERATING SYSTEM CONCEPTS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: 25BACSETC01

Course Description:

This laboratory course provides hands-on experience in operating system concepts through Unix/Linux commands, shell scripting, and C programming. Students implement and analyze CPU scheduling, process synchronization, deadlock avoidance, memory allocation, page replacement, file allocation, and disk scheduling algorithms, enabling them to develop practical skills in operating system resource management and performance analysis.

Course Objectives:

1. To understand the basics of Unix commands and shell programming.
2. To implement various CPU scheduling algorithms.
3. To implement IPC, Deadlock Avoidance and Detection Algorithms.
4. To implement Page Replacement Algorithms and memory allocation strategies
5. To implement File Allocation methods and Disk Scheduling algorithms

List of Programs:

1. A. Write basic UNIX commands to interact with UNIX operating systems.
   B. Write a shell script to print the biggest of n natural numbers.
   C. Write a shell script to print the factorial of first 'n' natural numbers.
   D. Write a shell script that counts the number of lines and words present in a given file.
   E. Write a shell script (small calculator) that adds, subtracts, multiplies and divides the given two integers. There are two division options: one returns the quotient and the other returns reminder. The script requires 3 arguments: The operation to be used and two integer numbers. The options are add (-a), subtract (-s), multiply (-m), quotient (-c) and reminder (-r).
2. Write C programs to implement the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)
3. (i) Write a C program to implement FCFS scheduling and calculate waiting time and turnaround time for the passengers arrive at a railway reservation counter in different times. The counter serves passengers in the same order they arrive.

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	8
P4	3	6

   (ii) Implement non-preemptive SJF scheduling for a photocopy center processes smaller copying jobs before larger jobs.

Process	Arrival Time	Burst Time
P1	0	7
P2	2	4
P3	4	1
P4	5	4

(iii) Write a C program to implement Priority Scheduling for Patients with higher emergency levels are treated first in a hospital.

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	2

(iv) Write a C program to implement Round Robin Scheduling for an online gaming server allocates equal CPU time to all active players using time-sharing.

Process	Burst Time
P1	24
P2	3
P3	3

4. Write a C program to implement mutual exclusion using semaphores for a banking application, multiple users try to access and update the same bank account balance simultaneously. To avoid data inconsistency, only one user should access the critical section at a time.
5. Write a C program for a restaurant kitchen prepares food orders while delivery agents collect prepared orders for delivery. The kitchen acts as the producer and delivery agents act as consumers.
6. Write a C program to implement Banker's Algorithm for deadlock avoidance that a bank server allocates limited resources such as database connections and processing units to multiple transactions and determine whether resource allocation is safe.

Allocation Matrix				Maximum Matrix			
Process	A	B	C	Process	A	B	C
P0	0	1	0	P0	7	5	3
P1	2	0	0	P1	3	2	2
P2	3	0	2	P2	9	0	2
P3	2	1	1	P3	2	2	2
P4	0	0	2	P4	4	3	3

7. Write C programs to implement First Fit, Best Fit and Worst Fit Memory Allocation for a cloud computing server has multiple free memory blocks of different sizes. Several user applications request memory for execution. The operating system must allocate memory using First Fit, Best Fit, and Worst Fit techniques and compare the allocation results. The program should display:
    (i) Allocated memory block for each process
    (ii) Unallocated processes (if any)
    (iii) Remaining memory after allocation
8. Write C programs to implement FIFO, LRU and LFU Page Replacement techniques for a smart digital library server manages limited memory frames to store frequently accessed e-books and documents. Multiple users request different pages from the server. Since the available memory frames are limited, the operating system must replace pages using different page replacement

algorithms. The program should display Page hits, Page faults, Frame status after each page request and total number of page faults.

9. Write C programs to implement Sequential, Indexed and Linked File Allocation for a university maintains a digital storage server for storing student records, assignments, and lecture materials. Files are stored in disk blocks using different file allocation methods. The program should display File name and allocated blocks, Starting block/index block, Linked list of blocks (for linked allocation) and Allocation status of each file.
10. Write C programs to implement disk scheduling algorithms FCFS (First Come First Serve), SSTF (Shortest Seek Time First), SCAN, C-SCAN for a cloud data center receives multiple disk access requests from users for reading and writing data. The program should display order of disk head movement, total seek time and average seek time.

COURSE OUTCOMES:

At the end of this course, students will be able to

CO1: Understand the use of Linux commands.
CO2: Compare the performance of processor scheduling algorithms.
CO3: Design algorithmic solutions for process synchronization problems.
CO4: Analyze the performance of various file management schemes.
CO5: Implement different page replacement and disk scheduling algorithms.

Text Books:

1. "Operating System Concepts", Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, 10th Edition, John Wiley and Sons Inc., 2018
2. "Modern Operating Systems", Andrew S Tanenbaum, Pearson, 5th Edition, Pearson India, 2024

Reference Books:

1. "Operating systems - A concept based Approach", D.M Dhamdhere, 2nd Edition, Tata McGraw-Hill, 2002.
2. "UNIX and Shell Programming", Behrouz A. Forouzan, Richard F. Gilberg, 1st Edition, Cengage India Private Limited, 2003.

Mode of Evaluation: Continuous Internal Evaluation of the Lab Experiments, Record, Viva-voce, and External Lab Examination

II Year II Semester

B. Tech II Year II Semester

25BAMATTC12 DISCRETE MATHEMATICAL STRUCTURES

L	T	P	C
3	0	0	3

Pre-requisite : Basic knowledge of set theory, elementary algebra, permutations, combinations and mathematical reasoning at the higher secondary level

Course Description:

This course introduces the concepts of discrete mathematical structures, which are essential for computer science and allied engineering. It covers logical reasoning, counting techniques, number theoretical systems, relations, recurrence relations, algebraic structures, and graph theory. The course emphasises mathematical modelling, abstraction, and problem-solving techniques widely used in algorithms, data structures, intelligent systems, and computer networks.

Course Objectives:

The objectives of this course are to enable students to:

1. Understand and apply propositional and predicate logic for formal reasoning to develop learning and intelligent systems.
2. Develop counting and number-theoretic techniques required for digital and security systems.
3. Analyse relations, order structures, and recurrence relations for computational problems.
4. Understand algebraic structures and modular arithmetic used in intelligent systems.
5. Apply graph and tree concepts to solve real-world, computational and networking problems.

UNIT I PROPOSITIONAL AND PREDICATE LOGICAL STRUCTURES 9 hours

Propositions and logical connectives, Tautologies and Contradictions, Logical equivalences, Quantifiers and predicates, Valid and invalid arguments, Theory of inferences, Rules of inference in propositional and predicate logic.

UNIT II COUNTING AND NUMBER THEORETICAL TECHNIQUES 9 hours

Counting finite sets, tuples, and lists. Divisibility, GCD-LCM and Euclidean algorithm, relatively prime integers, Euler's totient function and its properties, residue systems, Euler's and Fermat's theorems (without proof), linear congruences, the Chinese Remainder Theorem (without proof).

UNIT III ORDERED STRUCTURES AND RECURRENCE RELATIONS 9 hours

Equivalence relations, Equivalence classes, and partitions, Partial and Total Ordering, Lattices, Boolean Algebra. Recursively defined Functions**,** Recurrence Relations and their solutions using substitution and characteristic equation methods.

UNIT IV MODULAR ARITHMETIC AND ALGEBRAIC STRUCTURES 9 hours

Congruence classes and modular arithmetic, the structure of $\mathbb{Z}_n$, Units and zero divisors of $\mathbb{Z}_n$. Binary operations, Algebraic structures, groups, subgroups, cyclic groups, permutation groups, Rings and fields, Boolean rings.

UNIT V GRAPH AND TREE STRUCTURES 9 hours

Graphs, Representations of Graphs, Isomorphic Graphs, Paths and cycles, Eulerian and Hamiltonian Graphs, Graph Colouring. Trees, Spanning and Minimal Spanning Trees, Rooted Trees, Binary and Binary Search Trees.

Course Outcomes:

After successful completion of this course, the student will be able to:

CO1: Apply propositional and predicate logic, including rules of inference, to analyse the validity of mathematical and computational arguments.

CO2: Solve counting and number-theoretic problems using principles of combinatorics, divisibility,

modular arithmetic, and classical theorems such as Euler's and Fermat's theorems.

CO3: Analyse relations, equivalence classes, order structures, and solve recurrence relations arising in discrete and learning systems.

CO4: Examine and apply algebraic structures, including finite groups, Boolean rings, and finite fields, to model computational and intelligent systems.

CO5: Model and solve problems using graph and tree structures, including representations, traversals, and properties relevant to computing, designing and networking applications.

Text Books

1. Thomas Koshy, Discrete Mathematics with Applications, Elsevier, Academic Press, 1st edition, 2005.
2. Nadiya Gubareni, Introduction to Modern Algebra and Its Applications, CRC Press; 1st edition, 2020.

Reference Books

1. V. Ravichandran and Atul Kumar Razdan, Fundamental Discrete Structures, Springer Nature; 1st edition, 2025.
2. James L. Hein, Discrete Structures, Logic, and Computability, Jones and Bartlett Publishers, 4th edition, 2020.
3. Clifford Stein, Kenneth Bogart, Robert Drysdale, Discrete Mathematics for Computer Scientists, Pearson Education, 1st edition, 2011.
4. Thomas W. Hungerford, An Introduction to Abstract Algebra, Cengage India, 3rd edition, 2015.
5. Garry Haggard, John Schilpf, and Sue Whitesides, Discrete mathematics for Computer Science, Cengage learning, 1st edition, 2006.

Mode of Evaluation: Assignments, Mid Term Tests, End Semester Examination.

B. Tech II Year II Semester

25BACAMTC05 OBJECT ORIENTED PROGRAMMING CONCEPTS

L	T	P	C
3	0	0	3

Prerequisites: NIL

Course Description:

This course introduces fundamental object-oriented concepts and their implementation using Java. The course covers classes, objects, inheritance, polymorphism, abstraction, packages, interfaces, exception handling, multithreading, GUI programming, JDBC, and networking. It enables students to develop structured, reusable, and practical Java applications.

Course Objectives:

1. Understand the fundamentals of Object-Oriented Programming and Java programming concepts.
2. Develop Java programs using inheritance, packages, and interfaces.
3. Apply exception handling and multithreading concepts in Java applications.
4. Design simple GUI-based desktop applications using Swing, JavaFX, and event handling.
5. Develop database and networking applications using Java technologies.

UNIT I OBJECT ORIENTED THINKING AND JAVA BASICS 9 hours

Introduction to Object Oriented Programming, need for OOP paradigm, basic OOP concepts - classes, objects, encapsulation, inheritance, polymorphism, and abstraction. History and features of Java, Java buzzwords, data types, variables, scope and lifetime of variables, operators and expressions, control statements, arrays, type conversion and casting, simple Java programs, concepts of classes and objects, constructors and methods, method overloading, access modifiers, this keyword, basics of inheritance and method overriding, recursion, garbage collection, and overview of nested and inner classes.

UNIT II STRINGS, INHERITANCE, PACKAGES AND INTERFACES 9 hours

Strings: Strings, String Handling, Inheritance concepts, types of inheritance, use of super keyword, method overriding, dynamic method dispatch, abstract classes, packages, creation and usage of packages, interfaces, implementation of interfaces, multiple inheritance using interfaces, and development of reusable components.

UNIT III EXCEPTION HANDLING AND MULTITHREADING 9 hours

Exception handling fundamentals, try, catch, throw, throws, finally block, built-in exceptions, user-defined exceptions, multithreading concepts, thread life cycle, thread priorities, synchronization, and inter-thread communication.

UNIT IV MODERN GUI PROGRAMMING 9 hours

Introduction to GUI Programming, limitations of Applets, basics of Swing and JavaFX, GUI components such as Labels, Buttons, TextFields, and TextAreas, layout managers, event handling using listeners, creation of simple desktop applications, and introduction to Scene Builder.

UNIT V DATABASE CONNECTIVITY AND NETWORKING 9 hours

Introduction to JDBC architecture, database connectivity using JDBC, JDBC drivers, statements, Result Set, Prepared Statement, and handling database operations using Java. Basics of networking, Internet addressing, URL handling, socket programming, client-server communication.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Understand object-oriented programming concepts and develop basic Java applications.
CO2: Apply inheritance, packages, and interfaces in Java program development.
CO3: Implement exception handling and multithreading techniques in Java applications.
CO4: Develop simple GUI-based desktop applications using Swing, JavaFX, and event handling concepts.
CO5: Develop database and networking applications using JDBC and Java networking concepts.

Textbooks:

1. "Java: The Complete Reference", Herbert Schildt, 9th Edition, McGraw Hill Education, 2014.
2. "Programming with Java", E. Balagurusamy, 6th Edition, McGraw Hill Education, 2019.

Reference Books:

1. "Core Java Volume I – Fundamentals", Cay S. Horstmann, 11th Edition, Pearson Education, 2018.
2. "Java: How to Program", Paul Deitel and Harvey Deitel, 11th Edition, Pearson Education, 2018.
3. "Thinking in Java", Bruce Eckel, 4th Edition, Pearson Education, 2006.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech II Year II Semester

25BACAMTC06 MACHINE LEARNING TECHNIQUES

L	T	P	C
3	0	0	3

Prerequisites: Nil

Course Description:

This course introduces essential machine learning techniques for data analysis, prediction, classification, clustering, and decision-making. It covers supervised and unsupervised learning, feature selection, model evaluation, optimization, and ensemble methods. Emphasis is placed on understanding algorithmic principles and applying suitable learning techniques to solve engineering and real-world computational problems effectively.

Course Objectives:

1. Understand the basic concepts, paradigms, stages, and data handling methods used in Machine Learning.
2. Learn distance and similarity measures used in nearest neighbor-based classification and regression models.
3. Understand the principles of Decision Trees, Random Forests, and Bayes classification techniques.
4. Study linear classification models, support vector machines, regression methods, and neural network learning algorithms.
5. Understand clustering methods and unsupervised learning techniques for grouping and analyzing data patterns.

UNIT I INTRODUCTION TO MACHINE LEARNING 9 hours

Evolution of Machine Learning – Paradigms for ML – Learning by Rote – Learning by Induction – Reinforcement Learning – Types of Data – Matching – Stages in Machine Learning – Data Acquisition – Feature Engineering – Data Representation – Model Selection – Model Learning – Model Evaluation – Model Prediction – Search and Learning – Data Sets.

UNIT II NEAREST NEIGHBOR-BASED MODELS 9 hours

Introduction to Proximity Measures – Distance Measures – Non-metric Similarity Functions – Proximity Between Binary Patterns – Classification Algorithms Based on Distance Measures – K-Nearest Neighbor Classifier – Radius Distance Nearest Neighbor Algorithm – KNN Regression – Performance of Classifiers –Performance of Regression Algorithms.

UNIT III DECISION TREE MODELS & BAYES CLASSIFIER 9 hours

Decision Trees: Decision Trees for Classification – Impurity Measures – Properties – Regression Based on Decision Trees –Bias–Variance Trade-off – Random Forests for Classification and Regression. Bayes Classifier: Introduction to the Bayes Classifier – Bayes' Rule and Inference – The Bayes Classifier and its Optimality –Multi-class Classification – Class Conditional Independence – Naive Bayes Classifier (NBC).

UNIT IV LINEAR DISCRIMINANTS & NEURAL MODELS 9 hours

Introduction to Linear Discriminants – Linear Discriminants for Classification – Perceptron Classifier – Perceptron Learning Algorithm – Support Vector Machines (SVM) – Linearly Non-Separable Case – Non-linear SVM – Kernel Trick – Logistic Regression – Linear Regression – Multi-layer Perceptrons (MLPs) – Backpropagation for Training an MLP.

UNIT V CLUSTERING **9 hours**

Introduction to Clustering – Partitioning of Data – Matrix Factorization – Clustering of Patterns – Divisive Clustering – Agglomerative Clustering – Partitional Clustering – K-Means Clustering – Soft Partitioning – Soft Clustering – Fuzzy C-Means Clustering – Rough Clustering – Rough K-Means Algorithm – Expectation Maximization (EM) Based Clustering – Spectral Clustering.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Explain the stages of Machine Learning and interpret different data representation and model evaluation techniques.

CO2: Apply proximity measures and nearest neighbor algorithms for classification and regression problems.

CO3: Analyze Decision Tree and Naive Bayes models for solving classification and regression tasks.

CO4: Apply linear models, SVMs, and neural network techniques for machine learning applications.

CO5: Analyze clustering algorithms for identifying patterns and grouping datasets in unsupervised learning.

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Anantha narayana, Universities Press (India), 2024

Reference Books:

1. "Machine Learning", Tom M. Mitchell, Mc Graw- Hill Publication, 2017.
2. "Machine Learning in Action", Peter Harrington, Dream Tech, 2012.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech II Year II Semester

25BACAMTC07 DATABASE MANAGEMENT AND AI INTEGRATION

L	T	P	C
3	0	0	3

Prerequisites: Nil

Course Description:

This course introduces Database Management Systems with AI and ML integration. It covers database architecture, modeling, normalization, SQL, transactions, query optimization, warehousing, ETL, NoSQL, distributed databases, MongoDB, and Cassandra. Students explore Python-based database connectivity, AI-ready pipelines, vector databases, embeddings, RAG, LLMs, and Text-to-SQL, emphasizing scalability, security, privacy, governance, and ethical data management.

Course Objectives:

1. Understand the fundamental concepts of database management systems, data models, database architecture, and relational database design principles.
2. Develop efficient relational database schemas using normalization techniques and implement database applications using SQL and transaction management concepts.
3. Analyze transaction processing, concurrency control, query optimization, and data warehousing techniques for reliable database systems.
4. Explore NoSQL databases, distributed database concepts, and big data technologies for handling large-scale and AI-oriented datasets.
5. Integrate databases with Artificial Intelligence and Machine Learning frameworks to build intelligent, scalable, and secure data-driven applications using modern AI technologies such as vector databases, LLMs, and Retrieval-Augmented Generation (RAG) applications.

UNIT I INTRODUCTION TO DATABASE MANAGEMENT AND AI SYSTEMS 9 hours

Introduction to Database Management Systems, Characteristics and Advantages of DBMS, File System vs Database System, Database Users and Database Administrator (DBA), Database Architecture (Three-Schema Architecture), Data Models: Hierarchical, Network, Relational and Object-Oriented Models, Entity Relationship (ER) Model, ER Diagrams and Design Principles, Relational Model, Keys and Integrity Constraints, Structured vs Unstructured Data (Brief Introduction for AI Applications)

UNIT II RELATIONAL DATABASE DESIGN AND SQL 9 hours

Relational Algebra Operations, Functional Dependencies, Database Normalization (1NF, 2NF, 3NF, BCNF, 4NF), Decomposition and Lossless Join, SQL Fundamentals, SQL Commands: DDL, DML, DCL and TCL, SQL Operators and Functions, Joins and Nested Queries, Views and Indexes, Procedures and Triggers.

UNIT III TRANSACTION MANAGEMENT AND DATABASE PROCESSING 9 hours

Transaction Concept, ACID Properties, Transaction States, Concurrency Control Basics, Lock-Based Protocols, Deadlock and Recovery Concepts, Query Processing and Query Optimization, Data Cleaning and Preprocessing using SQL, Data Warehousing Concepts, OLTP vs OLAP Systems, ETL Process for Data Integration

UNIT IV NoSQL DATABASES AND BIG DATA FUNDAMENTALS 9 hours

Introduction to NoSQL Databases, Types of NoSQL Databases: Key-Value Stores, Document Databases, Column-Family Databases, Graph Databases. MongoDB Fundamentals, Cassandra Fundamentals, Data Modeling in NoSQL, CAP Theorem, Distributed Databases, SQL vs NoSQL Databases, Handling Large-Scale Data for AI Applications,

UNIT V DATABASE MANAGEMENT AND AI INTEGRATION 9 hours

Introduction to AI-Integrated Database Systems, Database Connectivity with Python (SQLite, MySQL, PostgreSQL), Data Access using Pandas and SQLAlchemy, Database APIs and ORM Concepts, AI Data Pipelines for Machine Learning Applications, Feature Engineering and Feature Stores, Integration of Databases with Machine Learning Frameworks (Scikit-learn, TensorFlow, PyTorch), Vector Databases and Embeddings (FAISS, ChromaDB, Pinecone – Introduction), Retrieval-Augmented Generation (RAG) Concepts, Large Language Models (LLMs) for Natural Language to SQL (Text-to-SQL), AI-assisted Query Optimization and Intelligent Database Monitoring, AI-powered Data Governance, Privacy, Security and Ethical Considerations, Case Studies on AI-Driven Database Applications.

Course Outcomes:

At the end of this course students will demonstrate the ability to

CO1: Explain the architecture, components, data models, and design principles of database management systems, and differentiate between traditional file systems and modern database systems.

CO2: Design normalized relational database schemas and implement database applications using SQL, including queries, views, indexes, stored procedures, and triggers.

CO3: Apply transaction management, concurrency control, query optimization, ETL processes, and data warehousing techniques to ensure efficient and reliable database processing.

CO4: Evaluate and implement NoSQL database solutions, distributed database concepts, and big data technologies suitable for AI-enabled applications.

CO5: Integrate relational and NoSQL databases with AI/ML frameworks, implement AI-ready data pipelines, utilize vector databases and Retrieval-Augmented Generation (RAG), and develop intelligent database-driven applications while addressing privacy, security, and ethical considerations.

Text Books:

1. "Database System Concepts", Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 7th Edition, McGraw Hill, 2019.
2. "Designing Data-Intensive Applications", Martin Kleppmann, O'Reilly Media, 2017.

Reference Books:

1. "MongoDB: The Definitive Guide", Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, 3rd Edition, O'Reilly Media, 2019
2. "Data Management for Researchers", Kristin Briney, Pelagic Publishing, 2020.

Mode of Evaluation: Assignments, Mid Term Tests and End Semester Examination.

B. Tech II Year II Semester

25BACAMLC04 OBJECT ORIENTED PROGRAMMING CONCEPTS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Nil

Course Description:

This course provides hands-on experience in developing Java applications using fundamental programming and object-oriented concepts. Students implement programs involving classes, inheritance, polymorphism, abstraction, interfaces, exception handling, multithreading, synchronization, collections, and graph algorithms through real-world problem scenarios.

Course Objectives:

1. To understand and implement basic Java programming constructs.
2. To apply object-oriented concepts such as inheritance, polymorphism, and abstraction.
3. To develop modular programs using classes, interfaces, and packages.
4. To implement exception handling, multithreading, and GUI applications in Java.
5. To solve real-world and discrete mathematics problems using Java programming.

List of Programs:

1. a. Design a JAVA program for a supermarket billing system to demonstrate primitive data types, variables, and arithmetic operations.
   b. Create a JAVA program for a railway reservation system using command line arguments and control statements to display ticket details.
2. a. Implement a JAVA program for a student record management system using arrays and string operations.
   b. Build a JAVA program for a library application to check palindrome book IDs and perform matrix addition for stock management.
3. a. Model a JAVA program for an employee payroll system using classes, objects, constructors, and methods.
   b. Simulate an online shopping application demonstrating constructor overloading and method overloading for different product categories.
4. a. Construct a JAVA program for a banking system using static members and this keyword to manage customer accounts.
   b. Illustrate a university management system implementing single and multilevel inheritance for students and faculty.
   c. Create a JAVA program for a transport management system illustrating method overriding and the use of super keyword.
5. Design a JAVA program for a digital payment gateway using abstract classes and interfaces for multiple payment methods.
6. Implement an online examination system demonstrating exception handling and user-defined exceptions for invalid operations.
7. Simulate a movie ticket booking system using Thread class and Runnable interface to handle multiple users simultaneously.
8. Develop an inventory management system demonstrating synchronization and Producer–Consumer problem for stock updates.

9. Create a JAVA program for a social networking platform to perform set operations such as union, intersection, and difference on user groups.
10. Implement a JAVA program for a city navigation system using adjacency matrix and graph algorithms like BFS, DFS, and Dijkstra's shortest path algorithm.

COURSE OUTCOMES:

At the end of this course, students will be able to

CO1: Write and execute Java programs using basic syntax, data types, and control statements.
CO2: Apply arrays, strings, and object-oriented concepts such as classes and objects in problem solving.
CO3: Implement inheritance, polymorphism, abstraction, interfaces, and packages in Java applications.
CO4: Develop programs using exception handling, multithreading, and GUI (AWT/Swing) concepts.
CO5: Apply Java programming to solve real-world problems including JDBC operations and discrete mathematics problems.

Text Books:

1. "Java: The Complete Reference", Herbert Schildt, 11th Edition, McGraw Hill Education, 2019.
2. "Programming with Java", E. Balagurusamy, 6th Edition, McGraw Hill Education, 2019.

Reference Books:

1. "Introduction to Java Programming", Y. Daniel Liang, 10th Edition, Pearson Education, 2015.
2. "Java How to Program", Paul Deitel and Harvey Deitel, 11th Edition, Pearson Education, 2018.
3. "The Java Language Specification", James Gosling, Bill Joy, Guy Steele and Gilad Bracha, 3rd Edition, Oracle Press, 2005.

Mode of Evaluation: Continuous Internal Evaluation of the Lab Experiments, Record, Viva-voce, and External Lab Examination

B. Tech II Year II Semester

25BACAMLC05 MACHINE LEARNING TECHNIQUES LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Nil

Course Description:

This course provides practical experience in implementing machine learning techniques for data analysis and prediction. Students perform data preprocessing, feature selection, classification, regression, clustering, model evaluation, and optimization using real-world datasets. The course emphasizes hands-on experimentation, algorithm comparison, performance analysis, and development of effective machine learning solutions using computational tool

Course Objectives:

1. To learn the methods of solving problems using Machine Learning.
2. To learn about computing central tendency measures and Data preprocessing techniques
3. To learn about classification algorithms
4. To learn about regression algorithms
5. To apply different clustering algorithms for a problem.

List of Programs:

1. A college wants to analyze and prepare student academic records collected from multiple departments for machine learning applications. Write a program to load datasets, perform data preprocessing techniques such as handling missing values, removing duplicates, encoding categorical data, and feature scaling. Merge related datasets containing student details and examination records, split the dataset into training and testing sets, and use Pandas methods such as describe(), head(), tail(), and info() to summarize and inspect the processed student performance data.
2. A retail company needs to visualize monthly sales trends and product performance. Write a program using Pandas built-in visualization functions to generate bar plots, histograms, line plots, and scatter plots for the sales dataset.
3. An e-commerce company wants to classify customers into different categories based on purchasing behavior. Apply the KNN classification algorithm to predict customer categories using a suitable dataset.
4. A real estate agency wants to estimate house prices based on features such as area, location, and number of rooms. Apply the KNN regression algorithm to predict house prices using an appropriate dataset.
5. A bank plans to automate loan approval decisions based on customer details. Demonstrate the Decision Tree classification algorithm and perform parameter tuning to improve prediction accuracy.
6. A healthcare organization wants to predict whether a patient is affected by a disease using medical parameters. Apply the Random Forest classification algorithm on a healthcare dataset to improve diagnosis accuracy.
7. A financial company aims to predict stock prices using historical market data. Apply the Random Forest regression algorithm and analyze the prediction performance.
8. An email service provider wants to classify emails as spam or non-spam based on email content. Implement the Naïve Bayes classification algorithm using a suitable dataset.
9. A telecom company wants to identify customers likely to discontinue services. Apply the Support Vector Machine (SVM) classification algorithm to predict customer churn.
10. A supermarket chain wants to group customers based on their purchasing patterns for targeted

marketing. Implement the K-Means clustering algorithm, evaluate clustering performance using Euclidean distance measures, and analyze the effect of different K values.

Course Outcomes:

At the end of this course, students will be able to

CO1: Practice various mathematical relationships within and across machine learning algorithms
CO2: Design and implement various machine learning algorithms in a range of real-world applications
CO3: Apply suitable machine learning techniques for data handling
CO4: Analyze the paradigms of supervised and un-supervised learning
CO5: Evaluate the performance of algorithms

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Anantha narayana, Universities Press (India), 2024

Reference Books:

1. "Machine Learning", Tom M. Mitchell, Mc Graw- Hill Publication, 2017.
2. "Machine Learning in Action", Peter Harrington, Dream Tech, 2012.

Mode of Evaluation: Continuous Internal Evaluation of the Lab Experiments, Record, Viva-voce, and External Lab Examination

B. Tech II Year I Semester

25BACAMLC06 DATABASE MANAGEMENT AND AI INTEGRATION LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Nil

Course Description:

This laboratory course provides practical experience in designing, implementing, and managing database systems through real-world applications. Students develop ER diagrams, relational schemas, normalization, SQL queries, transactions, views, and indexes. Hands-on exercises cover Python-based database connectivity using MySQL, SQLite, Pandas, and SQLAlchemy, along with data preprocessing, machine learning, AI-driven applications, and intelligent solutions across diverse domains.

Course Objectives:

1. Provide hands-on experience in designing and implementing database systems using real-world scenarios such as healthcare, university, banking, e-commerce, and HR systems.
2. Develop the ability to design ER diagrams, normalize databases, and create relational schemas with appropriate constraints.
3. Enable students to write and optimize SQL queries using joins, subqueries, aggregation, and transaction control.
4. Introduce database performance tuning techniques such as indexing, views, query optimization, database programming using Python and integration with MySQL/SQLite.
5. Bridge databases with AI/ML applications through data extraction, preprocessing, and model training and develop skills in building intelligent systems such as natural language to SQL conversion.

List of Programs:

1. **A hospital stores patient details, doctor assignments, and treatments**.
   a. Design ER diagram for Patient–Doctor–Treatment relationship
   b. Create relational tables with primary and foreign keys
   c. Insert sample data
   d. Perform basic SQL queries (SELECT, WHERE, ORDER BY)
2. **A university manages students, courses, and enrollments**.
   a. Identify candidate keys and foreign keys
   b. Create normalized schema (up to 3NF)
   c. Write SQL queries to fetch enrolled students per course
   d. Enforce integrity constraints
3. **An e-commerce platform tracks customers, orders, and products.**
   a. Perform INNER, LEFT, RIGHT JOIN operations
   b. Write nested queries to find:
      I. Customers with highest purchase value
      II. Products never ordered
4. **A library tracks books, members, and issue records.**
   a. Use GROUP BY and HAVING clauses
   b. Find most borrowed books
   c. Count overdue books per member category
5. **A bank handles fund transfers between accounts.**
   a. Simulate transactions using COMMIT and ROLLBACK
   b. Demonstrate ACID properties
   c. Handle failed transactions safely
   d.

6. **HR department manages employee salary and departments.**
   a. Create SQL views for department-wise payroll
   b. Create indexes for performance optimization
   c. Compare query execution time before and after indexing
7. **University stores student academic records in a database and uses AI techniques to recommend the most suitable professional elective course for each student based on their academic performance and interests.**
   a. Create a database containing student details, grades, skills, and available elective courses.
   b. Retrieve and preprocess student data from the database using Python (Pandas).
   c. Build a simple recommendation model (Rule-based or Machine Learning using Scikit-learn) to suggest appropriate elective courses.
   d. Store the recommended elective course back into the database and display the recommendations through SQL queries.
8. **A smart retail system updates inventory automatically using Python.**
   a. Connect MySQL/SQLite using Python (SQLAlchemy / connector)
   b. Perform CRUD operations using Python scripts
   c. Automate stock update based on AI-like input rules
9. **A company predicts customer churn using stored database data.**
   a. Fetch dataset from SQL database using Pandas
   b. Perform data cleaning and preprocessing
   c. Train a simple ML model (Scikit-learn) using database data
   d. Evaluate model accuracy
10. **Users ask questions in natural language, system converts them into SQL.**
    a. Accept user query like: *"Show top 5 highest salary employees"*
    b. Convert it into SQL using rule-based or LLM-based approach
    c. Execute query on database
    d. Display results dynamically

Course Outcomes:

At the end of this course, students will be able to

CO1: Design Entity-Relationship (ER) diagrams, convert them into relational schemas, and apply normalization techniques (up to 3NF) to develop efficient and consistent database systems.

CO2: Develop and execute SQL queries involving data retrieval, joins, nested queries, grouping, aggregation, transaction control commands, and demonstrate ACID properties in database applications.

CO3: Create and manage database objects such as views and indexes to improve query performance and maintain database efficiency.

CO4: Integrate databases with Python to perform CRUD operations, automate data handling tasks, and preprocess database data for machine learning applications.

CO5: Develop intelligent database applications by building simple predictive models using Python libraries and implementing natural language to SQL query conversion systems for dynamic data retrieval.

Text Books:

1. "Database System Concepts", Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 7th Edition, McGraw-Hill, 2019.
2. "Fundamentals of Database Systems", Ramez Elmasri and Shamkant B. Navathe, 7th Edition, Pearson, 2016.

3. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", Aurélien Géron, 2nd Edition, O'Reilly, 2019.

Reference Books:

1. "Database Management Systems", Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, McGraw-Hill, 2003.
2. "Mining of Massive Datasets", Jure Leskovec, Anand Rajaraman, and Jeffrey Ullman, 3rd Edition, Cambridge University Press, 2020.
3. "Designing Data-Intensive Applications", Martin Kleppmann, 1st Edition, O'Reilly, 2017.

Mode of Evaluation: Continuous Internal Evaluation of the Lab Experiments, Record, Viva-voce, and External Lab Examination

SKILL ENHANCEMENT COURSES

Skill Enhancement Course - I

25BACAMSC01 DATA SCIENCE FOR ENGINEERS

L	T	P	C
1	0	2	2

Prerequisites: Nil

Course Description:

This course introduces the fundamental concepts and practical applications of data science for engineering problems. It develops skills in data acquisition, preprocessing, visualization, statistical analysis, exploratory data analysis, and basic machine learning using Python-based tools. The course emphasizes hands-on analysis of engineering datasets and enables students to develop basic predictive analytics workflows while understanding ethical and responsible use of data.

Course Objectives:

1. To understand fundamental concepts of data science and its engineering applications.
2. To develop skills in data preprocessing, analysis, and visualization using Python tools.
3. To apply statistical and machine learning techniques for engineering data analysis.
4. To perform exploratory analysis and predictive modeling for engineering datasets.
5. To understand ethical and practical considerations in data-driven engineering solutions.

UNIT I INTRODUCTION TO DATA SCIENCE AND PYTHON TOOLS 6 hours

Introduction to Data Science, Data Science lifecycle, Types of engineering data, Data acquisition concepts, Python ecosystem for data science, NumPy fundamentals, Pandas basics, Data structures for analytics.

1. Install and configure Python data science environment and perform operations using NumPy and Pandas.
2. Load engineering datasets and perform basic data inspection and summary analysis.

UNIT II DATA PREPROCESSING AND VISUALIZATION 6 hours

Data cleaning, Handling missing values, Outlier detection basics, Data transformation, Data normalization, Feature preparation, Data visualization principles, Matplotlib and Seaborn basics, Visualization of engineering datasets.

1. Clean real-world datasets by handling missing values and inconsistent entries.
2. Create visualizations such as histograms, scatter plots, bar charts, and correlation heatmaps.

UNIT III STATISTICAL ANALYSIS AND EXPLORATORY DATA ANALYSIS 6 hours

Descriptive statistics, Probability fundamentals, Random variables, Probability distributions, Correlation analysis, Hypothesis testing basics, Exploratory Data Analysis (EDA), Interpretation of engineering datasets.

1. Perform exploratory data analysis and statistical summary computation on engineering datasets.
2. Analyze correlation and probability-based insights from experimental data.

UNIT IV MACHINE LEARNING FUNDAMENTALS FOR DATA SCIENCE 6 hours

Introduction to machine learning in data science, Supervised learning concepts, Regression basics, Classification basics, Model training and testing concepts, Performance evaluation metrics, Overfitting and model validation basics.

1. Build simple regression and classification models using engineering datasets.
2. Evaluate model performance using standard metrics.

UNIT V ENGINEERING DATA ANALYTICS APPLICATIONS AND ETHICS 6 hours

Predictive analytics for engineering applications, Time-series basics, Data-driven decision making, Engineering case studies, Ethical issues in data science, Bias and fairness, Data privacy fundamentals, Responsible use of engineering data.

1. Develop a mini predictive analytics workflow using an engineering dataset.
2. Analyze ethical challenges and propose responsible data usage strategies.

Course Outcomes:

At the end of this course students will demonstrate the ability to
CO1: Explain core concepts and lifecycle stages of MLOps systems.
CO2: Analyze data engineering and automation workflows in operational ML systems.
CO3: Evaluate model lifecycle management and CI/CD practices for ML deployment.
CO4: Assess deployment and monitoring strategies for scalable ML applications.
CO5: Examine governance, security, and advanced operational challenges in AI systems.

Text Books:

1. "Doing Data Science: Straight Talk from the Frontline", Cathy O'Neil, Rachel Schutt, 1st Edition, O'Reilly Media, 2013.
2. " Python Data Science Handbook", Jake VanderPlas, 1st Edition, O'Reilly Media, 2016.

Reference Books:

1. "Data Science from Scratch", Joel Grus, 2nd Edition, O'Reilly Media, 2019.
2. " The Elements of Statistical Learning ", Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2nd Edition, Springer, 2009.

Mode of Evaluation: Continuous Internal Evaluation , Assignments, Mid Term Tests and End Semester Examination.

Skill Enhancement Course - II

25BACAMSC02 FULL STACK DEVELOPMENT

L	T	P	C
1	0	2	2

Pre-requisite: Nil

Course Description:

This course introduces the fundamentals of full stack web application development through a progressive, hands-on approach. It covers web development fundamentals, HTML, CSS, JavaScript, server-side scripting, database connectivity, and frontend-backend integration. Students gain practical experience in developing interactive and dynamic web pages and culminate by designing and implementing a functional web application integrating frontend, backend, and database components.

Course Objectives:

1. To understand and create web pages using basic HTML5 tags, forms, tables, and CSS styling.
2. To apply CSS for page layout, colours, fonts, and simple responsive design.
3. To write basic JavaScript for form validation, DOM manipulation, and simple interactivity.
4. To understand how a web server works and create simple server-side scripts using Python.
5. To connect a web page to a Python script and display dynamic content.

UNIT I HTML5 BASICS AND WEB PAGE STRUCTURE 6 hours

Introduction to the Web: browsers, servers, HTTP. Structure of an HTML document: DOCTYPE, head, body. HTML tags: headings, paragraphs, lists, links, images, tables. HTML Forms: text, radio, checkbox, dropdown, submit. Semantic tags: header, nav, section, footer. Introduction to HTML attributes and comments.

1. Create a simple personal profile web page using headings, paragraphs, images, and links.
2. Build a registration form with text, radio buttons, checkboxes, dropdowns, and a submit button.

UNIT II CSS STYLING AND PAGE LAYOUT 6 hours

Introduction to CSS: inline, internal, and external styles. Selectors: tag, class, id. Box model: margin, border, padding, content. Typography: font-family, font-size, colour, text-align. Layout: display, float, basic Flexbox. CSS for forms and tables. Simple hover effects and transitions.

3. Style the registration form and profile page using an external CSS file.
4. Design a simple two-column webpage layout using Flexbox.

UNIT III JAVASCRIPT BASICS 6 hours

Introduction to JavaScript: variables, data types, operators. Conditional statements and loops. Functions: declaration, parameters, return values. DOM manipulation: getElementById, innerHTML, style changes. Events: onclick, onsubmit, onmouseover. Form validation: checking empty fields, email format, password match. Simple alert, prompt, and confirm dialogs.

5. Write a JavaScript function to validate a registration form before submission.
6. Create a simple image gallery where clicking a thumbnail shows the full image using DOM manipulation.

UNIT IV INTRODUCTION TO PYTHON FOR WEB 6 hours

What is a backend and why we need it. Introduction to Python: variables, input/output, conditions, loops, functions. Reading and writing text files in Python. Introduction to Python CGI: how a web page talks to Python. Creating a simple Python script that receives form data and prints a response. Running a local web server using Python.

7. Write a Python script that takes name and age as input and prints a greeting message.
8. Create an HTML form that submits name and email; write a Python script to read and display the submitted values.

UNIT V SIMPLE WEB PUBLISHING 6 hours

Combining HTML, CSS, and JavaScript into a complete web page. Linking an HTML form to a Python script using form action. Displaying form results on a new page. Reading and writing simple data to a text file using Python. Introduction to cookies: storing small data in the browser. Overview of how websites go live: domain, hosting, uploading files.

9. Create a feedback form in HTML and CSS; use a Python script to save the feedback to a text file.
10. Build a simple web page that reads saved feedback from a text file and displays it neatly using HTML.

Course Outcomes:

At the end of this course, students will be able to
CO1: Create well-structured web pages using HTML5 tags, forms, and tables.
CO2: Apply CSS to style and layout web pages effectively.
CO3: Write basic JavaScript for form validation and simple DOM interactions.
CO4: Create a simple Python script that receives and processes HTML form data.
CO5: Build a complete web page combining HTML, CSS, JavaScript, and a Python backend script.

Text Books:

1. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics", Jennifer Niederst Robbins, O'Reilly Media, 5th Edition, 2018.
2. "Flask Web Development: Developing Web Applications with Python", Miguel Grinberg, O'Reilly Media, 2nd Edition, 2018.

Reference Books:

1. "HTML and CSS: Design and Build Websites", Jon Duckett, John Wiley & Sons, 2011.
2. "Automate the Boring Stuff with Python", Al Sweigart, No Starch Press, 2nd Edition, 2019.
3. "JavaScript and JQuery: Interactive Front-End Web Development", Jon Duckett, Wiley, 2014.
4. "Introduction to the Web" — MDN Web Docs (developer.mozilla.org), Mozilla Foundation, freely available online.

Mode of Evaluation: Continuous Internal Evaluation , Assignments, Mid Term Tests and End Semester Examination.