MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(Deemed to be University)

MADANAPALLE

www.mits.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)

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)

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(Deemed to be University)

MADANAPALLE

B. Tech Four Year Curriculum Structure

Branch: COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)

Total Credits	160 Credits for 2025 (Regular) & 120 Credits for 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	25BACAITC01	Principles of Artificial Intelligence	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	25BACAITC03	Python Programming for AI	3	0	0	3	3
4	PCC	25BACAITC04	Design and Analysis of Algorithms	3	0	0	3	3
5	PCC	25BACAITC05	Fundamentals of Database Systems	3	0	0	3	3
6	PCC	25BACAILC01	Python Programming for AI Laboratory	0	0	2	2	1
7	PCC	25BACAILC02	Design and Analysis of Algorithms Laboratory	0	0	2	2	1
8	PCC	25BACAILC03	Database Systems 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**	**10**	**26**	**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	25BACAITC06	Object Oriented Programming using JAVA	3	0	0	3	3
3	PCC	25BACAITC07	Principles of Operating Systems	3	0	0	3	3
4	PCC	25BACAITC08	Machine Learning	3	0	0	3	3
5	OE		**Open Elective – I** *(Refer ANNEXURE - II)*	3	0	0	3	3
6	PCC	25BACAILC04	Object Oriented Programming using JAVA Laboratory	0	0	2	2	1
7	PCC	25BACAILC05	Operating Systems Laboratory	0	0	2	2	1
8	PCC	25BACAILC06	Machine Learning 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	PCC	25BACAITC09	Automata Theory and Compiler Design	3	0	0	3	3
2	PCC	25BACAITC10	Big Data Analytics and AI-Driven Application	3	0	0	3	3
3	PCC	25BACAITC11	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	25BACAILC07	Big Data Analytics and AI Application Laboratory	0	0	2	2	1
7	PCC	25BACAILC08	Deep Learning Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – III** *(Refer ANNEXURE - IV)*	1	0	2	3	2
9	INT	25BACAIIC01	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		Entrepreneurship Venture	3	0	0	3	3
2	PCC	25BACAITC12	Computer Networks	2	0	0	2	2
3	PCC	25BACAITC13	Computer Vision and AI Application	3	0	0	3	3
4	PCC	25BACAITC14	Natural Language Processing	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	25BACAILC09	Computer Vision and AI Application Laboratory	0	0	2	2	1
8	PCC	25BACAILC10	Natural Language Processing 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	HSMC		Organizational Behavior	3	0	0	3	3
2	PCC	25BACAITC15	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	25BACAILC11	Generative AI Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – IV** *(Refer ANNEXURE – IV)*	1	0	2	3	2
9	INT	25BACAIIC02	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	25BACAIPC01	Capstone Project	0	0	28	28	14
2	INT	25BACAIIC03	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.	25BACAIDM01	Introduction to Internet of Things
2.	25BACAIDM02	Cloud Computing
3.	25BACAIDM03	Human Computer Interaction
4.	25BACAIDM04	Human Computer Interaction (in Hindi)
5.	25BACAIDM05	Distributed Systems
6.	25BACAIDM06	Ethical Hacking
Any advanced Course offered by SWAYAM NPTEL can be appended in future.		

Discipline Elective – II		
S. No.	**Course Code**	**Course Title**
1.	25BACAIDC01	Reinforcement Learning
2.	25BACAIDC02	Stream Data Analytics
3.	25BACAIDC03	Dynamic Paradigms for IoT
4.	25BACAIDC04	UI and UX Design and Human-centered Design
5.	25BACAIDC05	Information Security Design
Any advanced courses can be appended in future.		

Discipline Elective – III		
S. No.	**Course Code**	**Course Title**
1.	25BACAIDC06	Explainable AI
2.	25BACAIDC07	Social Network Analytics
3.	25BACAIDC08	Smart IoT Architecture
4.	25BACAIDC09	Java Springboot
5.	25BACAIDC10	Privacy in Communication Network
Any advanced courses can be appended in future.		

Discipline Elective – IV		
S. No.	**Course Code**	**Course Title**
1.	25BACAIDC11	Agentic AI
2.	25BACAIDC12	Bioinformatics
3.	25BACAIDC13	Wearable Computing
4.	25BACAIDC14	DevOps
5.	25BACAIDC15	Web and IoT Security
Any advanced courses can be appended in future.		

Discipline Elective –V		
S. No.	**Course Code**	**Course Title**
1	25BACAIDC16	MLOps
2	25BACAIDC17	Edge AI
3	25BACAIDC18	IoT and Smart Cities
4	25BACAIDC19	Vibe Computing
5	25BACAIDC20	Cryptocurrency and Blockchain Technologies
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.	25BACAISC01	Full Stack Development - I
Any Courses can be appended in future.		

Skill Enhancement Course – II		
S. No.	**Course Code**	**Course Title**
1.	25BACAISC02	Full Stack Development - II
Any Courses can be appended in future.		

Skill Enhancement course – III		
Sl. No.	**Course Code**	**Course Title**
1.	25BACAISC03	Introduction to NoSQL with MongoDB
Any Courses can be appended in future.		

Skill Enhancement Course – IV		
Sl. No.	**Course Code**	**Course Title**
1.	25BACAISC04	Prompt Engineering
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

25BACAITC01 PRINCIPLES OF ARTIFICIAL INTELLIGENCE

L	T	P	C
3	0	0	3

Pre-requisite: None

Course Description:

This course is aimed to provide basic understanding of different intelligent agents in terms of Artificial Intelligence. This Course covers introduction to artificial intelligence, solving problems by various algorithms, Knowledge and Reasoning, Uncertain Knowledge and Reasoning.

Course Objectives:

1. The student should be made to study the concepts of Artificial Intelligence.
2. The student should be made to learn the methods of solving problems using Artificial Intelligence.
3. The student should be made to introduce the concepts of Expert Systems.
4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.
5. To learn different knowledge representation techniques.

UNIT I INTRODUCTION **9 hours**

Introduction: What is AI, Applications of AI, Foundation of AI, History of AI, A.I Representation, intelligent agents: Agents and Environments, the concept of rationality, Agent types and PEAS Descriptions.

UNIT II SEARCH ALGORITHMS **9 hours**

Problem solving: Well defined problems, Toy problems, Real world problems, Uninformed search strategies – Breadth First Search, Depth First Search, Search with Partial Information, senseless problems, Informed search strategies – Gready Best First Search, A* Search.

UNIT III LOGICS CONCEPTS **9 hours**

Logic concepts: First order logic. Inference in first order logic, Propositional vs. First Order Inference, Unification & Lifting Forward Chaining, Backward Chaining, Resolution.

UNIT IV KNOWLEDGE REPRESENTATION **9 hours**

Representation of Knowledge: Knowledge representation issues, Predicate Logic- Logic Programming, Semantic Nets- Frames and Inheritance.

UNIT V EXPERT SYSTEM ARCHITECTURE **9 hours**

Expert Systems: Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON: Expert systems shells.

Course Outcomes:

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

CO1: Understand the fundamental concepts of Artificial Intelligence.
CO2: Apply appropriate search strategy to solve complex problems.
CO3: Analyse the different logic concepts in AI
CO4: Infer knowledge using suitable knowledge representation schemes for complex problem.
CO5: Discuss the techniques used for building expert systems by integrating AI to solve real-world problems.

Text Books:

1. S. Russel and P. Norvig, “Artificial Intelligence – A Modern Approach”, SecondEdition, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, Mc Graw Hill

Reference Books:

1. David Poole, Alan Mackworth, Randy Goebel,”Computational Intelligence: a logical approach”, Oxford University Press.
2. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problemsolving”, Fourth Edition, Pearson Education.
3. J. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers.
4. Artificial Intelligence, SarojKaushik, CENGAGE Learning.

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.

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: -

Course Objectives:

The main objective of the course is to

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:

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

Online Learning Resources

1. Computer Architecture And OrganizationBy Prof. Indranil Sengupta, Prof. Kamalika Datta IIT Kharagpur
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_cs88
2. Computer Architecture, Instructor: David Wentzlaff
   https://www.coursera.org/learn/comparch

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

B. Tech II Year I Semester

25BACAITC03 PYTHON PROGRAMMING FOR AI

L	T	P	C
3	0	0	3

Course Objectives:

The main objective of the course is to

1. Understand the fundamental concepts of Python programming language.
2. Explore the data structures and functions used in Python.
3. Expose the Object Oriented programming paradigm in Python with important libraries.
4. Understand the importance of Numpy in association with AI.
5. Analyse data using Pandas can understand patterns using live applications

UNIT I PYTHON PROGRAMMING FUNDAMENTALS 9 hours

Introduction to Python and AI programming environment - Python installation in different environment, Variables, constants, data types, type conversion - Operators and expressions - Input and output statements - Conditional statements: if, if-else, nested if, elif - Looping statements: for, while - Loop control statements: break, continue, pass - Strings and string operations - Basic problem-solving problems.

UNIT II DATA STRUCTURES AND FILES IN PYTHON 9 hours

Lists, tuples, sets, and dictionaries - List, Set, Dictionary Comprehensions - List, Set, Dictionary operations for data representation - List, Set, Dictionary methods - Modules and packages.

UNIT III CLASSES, OBJECTS AND FUNCTIONS IN PYTHON 9 hours

Principles of object-oriented programming-Classes and objects - Constructors and destructors - Instance variables and class variables - Inheritance - Polymorphism - Encapsulation - Method overriding. Functions, Recursive, Lambda, Built-in functions: map(), filter(), reduce(), zip(), enumerate().

UNIT IV NUMERICAL COMPUTING WITH NUMPY 9 hours

Numpy Introduction - Creating and Manipulating Arrays - Indexing & Slicing - Array Attributes & Operations - Sorting and Manipulation - Handling missing data - Matrix operations - Set operations - Random number -generation - Statistical Functions.

UNIT V DATA ANALYTICS USING PYTHON 9 hours

Pandas Introduction - Pandas Series and DataFrame - Reading CSV, Excel, and JSON datasets - Data cleaning and missing value handling - Data filtering, grouping, sorting, and aggregation - Data normalization and standardization - Data visualization using - Matplotlib and Seaborn - Correlation analysis - Exploratory Data Analysis.

Course Outcomes:

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

CO1: Understand basics and fundamental bulding blocks of Python Programming language.
CO2: Organize, process, and manipulate structured datasets using Python data structures and functional paradigms.
CO3: Analyze the OOPs concepts in Python and its library support towards Artificial Intelligence.
CO4: Perform Numerical operations using Numpy Concepts also can discover patterns, trends, anomalies, and insights from data.
CO5: Organize Data using Pandas data analytic concepts also can solve domain-specific problems.

Text Books:

1. Dr.R. Nageswara Rao (Author), Core Python Programming, 2nd Edition.
2. Mr Mark Summerfield, Programming in Python 3, 2nd Edition 2009.
3. Mr Mark Lutz, Learning Python, O'Reilly Media, 5th Edition-2013.

Reference Books:

1. Python Crash Course, Mr Eric Matthes, Starch Press, 3rd Edition 2023.
2. Artificial Intelligence: A Modern Approach, Stuart J. Russell, Peter Norvig, Pearson Education, 4th edition 2022
3. Python for Data Analysis, Mr Wes McKinney, O'Reilly, 3rd Edition 2022.

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

B. Tech II Year I Semester

25BACAITC04 DESIGN AND ANALYSIS OF ALGORITHMS

L	T	P	C
3	0	0	3

Pre-requisite: **Data Structures and Algorithms**

Course Objectives:

The main objective of the course is to

1. Mathematical approach for Analysis of Algorithms.
2. Methods and techniques for analyzing the correctness and resource requirements of algorithms.
3. Different paradigms of algorithm design including recursive algorithms, divide-and-conquer algorithms, dynamic programming, greedy algorithms, Backtracking, Branch and Bound and graph algorithms.
4. To introduce Complexity Theory with NP and Approximation.
5. To examine the chromatic number decision problem in NP complete problem.

UNIT I INTRODUCTION & DIVIDE AND CONQUER **9 hours**

Algorithm: Pseudo code for expressing algorithms; Performance analysis: Space complexity, time complexity; Asymptotic notations: Big O notation, omega notation, theta notation and little o notation, amortized complexity;

Divide and Conquer: General method, binary search, quick sort, merge sort, Stassen's matrix multiplication.

UNIT II SEARCHING AND TRAVERSAL TECHNIQUES **9 hours**

Disjoint set operations, union and find algorithms; Efficient non-recursive binary tree traversal algorithms, spanning trees;

Graph traversals: Breadth first search, depth first search, connected components, biconnected components.

UNIT III GREEDY METHOD AND DYNAMIC PROGRAMMING **9 hours**

Greedy method: The general method, job sequencing with deadlines, knapsack problem, minimum cost spanning trees, single source shortest paths.

Dynamic programming: The general method, matrix chain multiplication optimal binary search trees, 0/1 knapsack problem, single source shortest paths, all pairs shortest paths problem, the travelling salesperson problem.

UNIT IV BACKTRACKING AND BRANCH AND BOUND **9 hours**

Backtracking: The general method, the 8 queen's problem, sum of subsets problem, graph coloring, Hamiltonian cycles;

Branch and bound: The general method, 0/1 knapsack problem, least cost branch and bound solution, first in first out branch and bound solution, travelling salesperson problem.

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

Basic concepts: non-deterministic algorithms, the classes NP - Hard and NP, NP Hard problems, clique decision problem, chromatic number decision problem, Cook's theorem.

Course Outcomes:

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

CO1: Find the (worst case, randomized, amortized) running time and space complexity of given algorithms using techniques such as recurrences and properties of probability.

CO2: Make Use of appropriate tree traversal techniques for finding shortest path.

CO3: Compare Identify suitable problem-solving techniques for a given problem and finding optimized solutions using Greedy and Dynamic Programming techniques

CO4: Apply greedy algorithm Utilize backtracking and branch and bound techniques to deal with traceable and in-traceable problems.

CO5: Apply Describe the classes P, NP, NP-Hard, and NP- complete for solving deterministic and non-deterministic problems.

Text Books:

1. Ellis Horowitz, Satraj Sahni, Sanguthevar Rajasekharan, "Fundamentals of Computer Algorithms", Universities Press, 2nd edition, 2015.
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D, "The Design and Analysis of Computer Algorithms", Pearson India, 1st edition, 2013.
3. Micheal T. Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet examples", Second Edition, Wiley Publication, 2006

Reference Books:

1. Levitin A, "Introduction to the Design and Analysis of Algorithms", Pearson Education, 3rd edition, 2012.
2. Goodrich, M. T. R Tamassia, "Algorithm Design Foundations Analysis and Internet Examples", John Wileyn and Sons, 1st edition, 2001.
3. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Third Edition, Pearson Education, 2012

Online Learning Resources

1. https://nptel.ac.in/courses/106101059

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

B. Tech II Year I Semester

25BACAITC05 FUNDAMENTALS OF DATABASE SYSTEMS

L	T	P	C
3	0	0	3

Course Objectives:

1. To understand the concept of DBMS and ER Modelling.
2. To explain the normalization, Query optimization and relational algebra.
3. To have an introductory knowledge about the storage and query processing techniques and the basic concepts of Information retrieval techniques
4. To learn about the internal storage structures using different file and indexing techniques which will help in physical DB design
5. To apply the concurrency control, recovery, security and indexing for the real time data.

UNIT I DATABASE SYSTEMS CONCEPTS AND DATA MODELING 9 hours

Introduction to Databases- File System Vs Database System - Data Models- Schemas and Instances - DBMS Architecture - Database Applications, Types of Databases.

Entity Relationship Model: Types of Attributes, Entities and Entity set, Relationship, Structural Constraints - Relational Model, Relational model Constraints - Mapping ER model to a relational schema - Specialization and generalization using ER Diagrams.

UNIT II SQL 9 hours

Introduction to SQL – Basic SQL queries – Queries involving multiple tables – SQL functions: aggregate functions, string functions, date and time functions, NULL value handling functions – Subqueries – Integrity constraints – Views and indexes – Triggers – Dynamic SQL – Cursors – Introduction to JDBC – Stored Procedures.

UNIT III NORMALIZATION 9 hours

Translating SQL Queries into Relational Algebra and Relational Calculus, Guidelines for Relational Schema – Functional dependency; Normalization, Boyce Codd Normal Form, Multi-valued dependency and Fourth Normal form; Join dependency and Fifth Normal form.

UNIT IV DATA STORAGE AND TRANSACTION MANAGEMENT 9 hours

Storage strategies: File organization, indices, B-trees, B+ -trees, hashing.

Transaction concepts – ACID properties – Transaction schedules – Serializability – Concurrency Control – Two-Phase Locking Techniques – Timestamp-based Concurrency Control – Recovery Concepts – Recovery techniques based on deferred update and immediate update – Buffer management

UNIT V DATABASE SECURITY AND RECENT TRENDS 9 hours

Database Authentication, Authorization and access control, DAC, MAC and RBAC models, SQL injection. Introduction, Need of NoSQL, CAP Theorem and Recent trends.

Course Outcomes:

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

CO1: Construct an ER model and derive the relational schemas from the model.
CO2: Understand the conceptual and logical database design using SQL queries.
CO3: Apply Normalization to improve database design.
CO4: Interpret the basic issues of storage and transaction management.
CO5: Analyse the fundamental security concepts for database.

Text Books:

1. A. Silberschatz, H. F. Korth S. Sudershan, Database System Concepts, McGraw Hill, 7th Edition 2021
2. R. Elmasri S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 2015.

Reference Books:

1. Raghu Ramakrishnan, Database Management Systems, Mcgraw-Hill,4th edition,2015.
2. Thomas Connolly, Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation and Management,6th Edition,2012.

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

B. Tech II Year I Semester

25BACAILC01 PYTHON PROGRAMMING FOR AI LABORATORY

L	T	P	C
0	0	3	1.5

Course Objectives:

1. To gain hands on experience in basics of python programming language.
2. To demonstrate the data structures and functions in Python
3. To explore data analysis and visualization packages in Python
4. To understand rule based AI using Python

List of Experiments:

Using Python environment

1. Write a Python code to create a calculator that performs addition, subtraction, multiplication, and division based on user choice.
2. Write a Python code to accept marks from the user and display the grade.
3. Write a Python code to generate multiplication tables for a given number.
4. Write a Python code to store and manage student records using dictionaries and lists.
5. Write a Python code to accept a list of numbers and perform multiple operations a.Find the sum of squares of all the numbers. b. Display largest and Smallest numbers.
6. A university wants to automate how student attendance is logged for various courses. Instead of tracking everything via loose dictionaries, they need a structured system where a Course can dynamically enroll and log attendance for multiple Student objects.
7. Write a Python code to develop a payroll system that stores employee salary details in CSV format.Perform the following operations a.Calculate Salary with allowances b. Read and Write CSV files.
8. Write a Python code to create a mini library management application. Perform the following tasks a. Add books b. Issue and return books c. Avoid duplicate entries using sets.
9. Write a Python code to create a student management system to store and display student details.
10. Write a Python code to create an employee payroll management system using Pandas.
11. Write a Python code to build a scientific calculator using NumPy functions. a.Matrix operations b.Mean,median calculations.
12. Write a Python code to create charts and graphs for student or sales data. a.Line chart b.Bar graph c.Pie chart d.Plot data from CSV file
13. Write a Python code to develop a simple expert system that predicts a possible illness based on symptoms entered by the user. a.Accept symptoms like fever, cough, headache b. Use rules to identify diseases c. Display diagnosis and suggestions
14. Write a Python code to create a rule-based AI system that suggests career options based on student interests and marks. a.Input marks and interests b.Suggest careers (Engineering, Medicine, Arts, Business).
15. Write a Python code to build a chatbot that answers basic college-related questions. a.Respond to user queries:Course details, Fee structure, Timings, Admissions.

Hardware Requirements:

- **Processor:** Intel Core i5/i7 or AMD Ryzen 5/7
- **RAM:** 8 GB or higher
- **Storage:** SSD with 10 GB free space
- **Graphics:** Integrated or dedicated graphics card (optional for AI/Data Visualization tasks)
- **Internet Connection:** Required for installing libraries, updates, and online IDE access

Software requirements:

- **Python Interpreter:** Python 3.x
- **IDE/Editor:** PyCharm, Visual Studio Code, Jupyter Notebook, or IDLE
- **Libraries/Packages:** NumPy, Pandas, Matplotlib, Scikit-learn (depending on application)
- **Package Manager:** pip or Anaconda

Course Outcomes:

CO1: To explore the basic concepts in Python programming language.

CO2: To Create small applications using python functions and packages.

CO3: Analyze data using numpy and panda concepts.

CO4: To handle the exception using python in built libraries.

CO5: Generate graphs and charts for visualizing the data using python libraries.

Reference Books:

1. Programming and Problem Solving with Python, Ashok Namdev Kamthane, Amit Ashok Kamthane, Pearson Education India, First Edition 2018.
2. Data Analysis Using Python, Bharti Motwani, BPB Publications, Edition 2018.
3. Think Python, Allen B. Downey, O'Reilly, 2nd Edition 2015

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech II Year I Semester

25BACAILC02. DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: 25BACSELC03

Course Objectives:

The objectives of the course are to

1. The selection of Algorithmic technique and Data structures required for efficient development of technical and engineering applications.
2. The algorithmic design paradigms and methods for identifying solutions of optimization problems.
3. Implementation of different algorithms for the similar problems to compare their performance.

List of Experiments:

1. Implement Quick Sort and Merge Sort and observe the execution time for various input sizes (Average, Worst, and Best cases).
2. Implement the Knapsack Problem using the Greedy method.
3. Construct Minimum Spanning Tree using
4. Prim's Method (b) Kruskal's Method
5. Implement All Pairs Shortest Paths using Dynamic Programming method
6. Implement Single Source Shortest Paths using Dynamic Programming method
7. Write a program to solve 0/1 Knapsack problem Using Dynamic Programming.
8. Implement the Travelling Salesperson problem using Dynamic Programming.
9. Implement N-Queens Problem Using Backtracking Strategy.
10. Implement the Backtracking method to solve the Graph Coloring problem.
11. Design and implement sum of subsets problem using backtracking.
12. From a given vertex in a weighted connected graph, find shortest paths to other vertices
13. using Dijkstra's algorithm.
14. Design and implement in Java to find all Hamiltonian Cycles in a connected undirected
15. Graph G of n vertices using backtracking principle.
16. Design and implement to find a subset of a given set S = {Sl, S2,....., Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.
17. Implement any scheme to find the optimal solution for the Traveling Sales Person problem
18. and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.

Course Outcomes:

After completion of the course, Students will be able to

CO1: Apply Divide and conquer strategy to organize the data in ascending or descending order

CO2: Make use of Algorithmic Design paradigms to determine shortest distance and transitive closure of Directed or Undirected Graphs

CO3: Compare the efficiencies of traversal problems using different Tree and Graph traversal algorithms and examine branch and bound approach for solving Combinatorial optimization problems.

Reference Books:

1. Levitin A, "Introduction to the Design and Analysis of Algorithms", Pearson Education, 2008.
2. Goodrich, M.T. R Tomassia, "Algorithm Design foundations Analysis and Internet Examples", John Wiley and Sons, 2006.
3. Base Sara, Allen Van Gelder, "Computer Algorithms Introduction to Design and Analysis", Pearson, 3rd edition, 1999.

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

B. Tech II Year I Semester

25BACAILC03 DATABASE SYSTEMS LOBORATORY

L	T	P	C
0	0	3	1.5

Course Objectives:

1. To understand the components of DBMS and to study the database design.
2. To study the retrieval of data using relational algebra and calculus and the concept of normal forms in the design of database.
3. To comprehend the structure of SQL Queries to query, update, and manage a database.
4. To understand all constraints to develop a business application using cursors, triggers and stored procedures.
5. To provide sufficient skill to utilize the DBMS concept in real time applications.

List of Experiments:

1. Design Conceptual database schema using ER Modelling Software Tools.
2. Development of Relational Database Schemas for Deposit/Customer/ borrow/ branch using DDL Constructs of SQL.
3. To perform various data manipulation commands such as select, insert, update etc. of SQL on Relational Database.
4. To perform various DCL and TCL construct of SQL on Relational Database.
5. Implement different types of referential and integrity constraints on Relation Database.
6. To apply SQL functions: aggregate functions, string functions, date and time functions, and NULL value handling functions.
7. To retrieve the queries using Group by, Having and Order by clauses of SQL.
8. Design and development of Banking database and perform various type of JOIN operations.
9. Insert the Data into table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL
10. Write a trigger that automatically deletes students when they graduate
11. Develop programs using features parameters in a CURSOR for UPDATE
12. a) Create a cursor to update the salary of employees in EMP table.
    b) Write a PL/SQL program to raise an Exception when the bonus exceeds salary.
13. Design and implementation real time project with database connection.

Course Outcomes:

CO1: Perform table creation, maintain, and manipulate a relational database using SQL.
CO2: Implement complex queries using SQL.
CO3: Apply Queries using Advanced Concepts of SQL.
CO4: Build PL/SQL programs including stored procedures, functions, cursors and triggers.
CO5: Develop a real-world application to access and render data.

Reference Books:

1. A. Silberschatz, H. F. Korth S. Sudershan, Database System Concepts, McGraw Hill, 7th Edition 2021.
2. R. Elmasri S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 2015.
3. Raghu Ramakrishnan, Database Management Systems,Mcgraw-Hill,4th edition,2015.
4. Thomas Connolly, Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation and Management,6th Edition,2012.

Mode of Evaluation: Continuous Internal Evaluation and End Semester 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

25BACAITC06 OBJECT ORIENTED PROGRAMMING USING JAVA

L	T	P	C
3	0	0	3

Course Objectives:

The learning objectives of this course are to:

1. To learn Java programming fundamentals including data types, type casting, tokens, control statements, looping statements, and user input handling.
2. To understand Java programs using classes, objects, constructors, methods, and access control mechanisms.
3. To Learn how to extend Java classes with inheritance and interfaces in Java applications.
4. To understand string handling techniques, package creation, and exception handling mechanisms in Java programming.
5. To understand Java I/O stream classes for file handling operations and to design multithreaded applications using Java threads.

UNIT I **9 hours**

Object Oriented Programming: Basic concepts, Program Structure in Java: Introduction, Writing Simple Java Programs, Data types, Type casting, Tokens in Java Programs, Java Statements –Control statements, Looping Statements, Break Statement, Continue Statement. Command Line Arguments, User Input to Programs, Escape Sequences, Comments, Programming Style.

UNIT II **9 hours**

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of a Class, Constructors, Constructor Overloading, Nested Classes, Final Classes and Methods, Passing Primitive Data Types and Objects as Arguments, this and static Keywords.

Methods: Introduction, Defining Methods, Method Overloading, Class Objects as Parameters in Methods, Access Control, Recursive Methods, and Nesting of Methods.

UNIT III **9 hours**

Arrays: One-dimensional and Multidimensional Arrays.

Inheritance: Basics of Inheritance, Types of Inheritance, Usage of super Keyword, Method Overriding, Abstract Classes, and Final Keyword.

Interfaces: Creating, Implementing, Using, Extending, and Nesting Interfaces.

UNIT IV **9 hours**

String Handling in Java: Introduction, CharSequence Interface, String Class, Methods for Extracting Characters from Strings, String Comparison, Modifying and Searching Strings, StringBuffer Class, and Wrapper Classes.

Packages: Defining, Finding, and Importing Packages, Member Access.

Exception Handling: Fundamentals of Exception Handling, Types of Exceptions, Multiple Catch Blocks, Nested Try Blocks, throw and throws Keywords, finally Block, Built-in Exceptions, and User-defined Exceptions.

UNIT V **9 hours**

I/O Streams: Byte Stream Classes and Character Stream Classes.

Multithreaded Programming: Introduction, Need for Multiple Threads, Multithreaded Programming for Multi-core Processors, Thread Class, Main Thread and Creation of New Threads, Thread States, Thread Lifecycle, Synchronization, and Inter-thread Communication.

Course Outcomes:

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

CO1: Understanding of Java program structure, data types, type casting, control flow statements, and user input handling.

CO2: Explain the concepts of classes, objects, constructors, methods, access control, and keywords used in Java programming.

CO3: Implement arrays, inheritance and interfaces to develop reusable, modular, and well-structured Java programs.

CO4: Apply string handling, packages, and exception handling concepts in developing Java applications.

CO5: Analyze and implement I/O streams and multithreading concepts for efficient file handling and concurrent programming in Java.

Text Books:

1. Java: One Step Ahead, Anita Seth and B. L. Juneja, Oxford University Press, First Edition, 2017.
2. Joy with JAVA, Fundamentals of Object Oriented Programming, DebasisSamanta, MonalisaSarma, Cambridge, 2023.

Reference Books:

1. The complete Reference Java, 11thedition, Herbert Schildt,TMH.
2. Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson
3. JAVA for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

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

B. Tech II Year II Semester

25BACAITC07 PRINCIPLES OF OPERATING SYSTEMS

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Objectives:

The main objectives of this course are to

1. To understand the fundamental concepts and techniques of Operating Systems.
2. To study the concepts of linux shell and scheduling.
3. To understand the concepts in deadlocks and process management.
4. To understand the concepts in memory managements and IPC mechanism.
5. To study the concept of file system concepts and sockets.

UNIT I INTRODUCTION TO OPERATING SYSTEMS 9 hours

Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Free and Open-Source Operating Systems, System Structures: Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Building and Booting an Operating System, Operating system debugging

UNIT II PROCESS AND CPU SCHEDULING 9 hours

Processes: Process Concept, Process scheduling, Operations on processes, Inter-process communication. Threads and Concurrency: Multithreading models, Thread libraries, Threading issues. CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling

UNIT III SYNCHRONIZATION AND DEADLOCKS 9 hours

Synchronization Tools: The Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores, Monitors, Classic problems of Synchronization. Deadlocks: system Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from Deadlock

UNIT IV MEMORY MANAGEMENT AND VIRTUAL MEMORY 9 hours

Memory-Management Strategies: Introduction, Contiguous memory allocation, Paging, Structure of the Page Table, Swapping. Virtual Memory Management: Introduction, Demand paging, Copy-on-write, Page replacement, Allocation of frames, Thrashing Storage Management: Overview of Mass Storage Structure, HDD Scheduling.

UNIT V FILE SYSTEM INTERFACE AND FILE PROTECTION 9 hours

File System: File System Interface: File concept, Access methods, Directory Structure; File system Implementation: File-system structure, File-system Operations, Directory implementation, Allocation method, Free space management; File-System Internals: File-System Mounting, Partitions and Mounting, File Sharing. Protection: Goals of protection, Principles of protection, Protection Rings, Domain of protection, Access matrix.

Course Outcomes:

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

CO1: Describe the basics of the operating systems, mechanisms of OS to handle processes, threads, and their communication.

CO2: Make use of process scheduling algorithms and synchronization techniques to achieve better performance of a computer system

CO3: Illustrate different conditions for deadlock and their possible solutions.

CO4: Analyse the memory management and its allocation policies. CO5: Implement the file protection techniques.

Text Books:

1. "Operating System Concepts", Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, 10th 2. Edition, John Wiley and Sons Inc., 2018.
2. "Modern Operating Systems", Andrew S Tanenbaum, Pearson, 5th Edition, 2022 New Delhi.

Reference Books:

1. "Operating Systems – A Spiral Approach", Ramaz Elmasri, A. Gil Carrick, David Levine, Tata McGraw Hill Edition, 2010.
2. "Operating Systems: Internals and Design Principles", William Stallings, 7th Edition, Prentice Hall, 2018.
3. "Operating Systems", Achyut S.Godbole, Atul Kahate, McGraw Hill Education, 2016.

Online Learning Resources

1. Operating Systems Course for Beginners
   https://www.youtube.com/watch?v=yK1uBHPdp30
2. Operating Systems Basics
   https://www.netacad.com/courses/operating-systems-basics?courseLang=en-US
3. Popular Operating System Courses and Certifications
   https://www.coursera.org/courses?query=operating%20system
4. NPTEL Operating System Fundamentals, IIT Kharagpur, Prof. Santanu Chattopadhyay
   https://nptel.ac.in/courses/106105214

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

B. Tech II Year II Semester

25BACAITC08 MACHINE LEARNING

L	T	P	C
3	0	0	3

Pre-requisite: None

Course Objectives:

The main objectives of this course are to:

1. Introduce the basic principles, learning approaches, and mathematical foundations of machine learning
2. help students understand how data is prepared and used to build regression and classification models
3. provide knowledge of regularization, generalization and model evaluation methods
4. introduce decision trees, support vector machines, and ensemble learning techniques for prediction tasks
5. familiarize students with clustering and dimensionality reduction methods used for discovering patterns in data

UNIT I MACHINE LEARNING PRELIMINARIES 9 hours

Introduction to ML: Machine Learning vs Traditional AI, Learning from data: features, labels, Types of learning: Supervised, Unsupervised, Semi-supervised, Self-supervised and Reinforcement, Applications of ML, Machine Learning lifecycle.

Mathematical foundations for ML: Basic probability rules, Conditional probability, Bayes' theorem, Random variables, Mean, variance, standard deviation, covariance and correlation.

Basic ML concepts: hypothesis, model parameters, loss function, training and testing, basic idea of optimization, and gradient descent.

UNIT II REGRESSION MODELS & REGULARIZATION 9 hours

Regression Fundamentals: Regression vs classification, simple linear regression, multiple linear regression, least squares estimation, cost function, gradient descent for regression, polynomial regression, underfitting, and overfitting.

Regularization and data preprocessing: bias-variance trade-off, Ridge regression, Lasso regression, feature selection using regularization, data distribution, noisy data, outliers, missing values, feature scaling, normalization, standardization, categorical encoding, and train-validation-test split. Regression evaluation: MAE, MSE, RMSE, and R^2 score.

UNIT III CLASSIFICATION MODELS 9 hours

Probabilistic Models: Bayesian decision theory, likelihood, prior probability, posterior probability, Maximum Likelihood Estimation, Maximum A Posteriori estimation and Gaussian Naïve Bayes classifier.

Classification Models: Logistic Regression, sigmoid function, decision boundary, multiclass classification, k-Nearest Neighbor for classification and regression, distance and similarity measures such as Euclidean distance, Manhattan distance, Minkowski distance and cosine similarity. Limitations of k-Nearest Neighbor.

UNIT IV DECISION TREES, SVM & MODEL EVALUATION 9 hours

Decision Tree Models: Decision trees for classification and regression, Impurity measures: Gini, entropy, information gain, overfitting in trees and pruning. Support Vector Machines: Intuition of maximum margin classifier, Types – linear, nonlinear, kernel, polynomial and SVR.

Model Evaluation Metrics: Confusion matrix, Accuracy, Precision, Recall, F1-score, ROC curve,

AUC, error analysis, class imbalance, k-fold cross-validation and hyperparameter tuning using grid search and random search.

UNIT V ENSEMBLE TECHNIQUES, UNSUPERVISED LEARNING & DIMENSIONALITY REDUCTION **9 hours**

Ensemble learning: combining multiple learners, model combination schemes, voting, bagging, boosting, stacking, Random Forest and AdaBoost.

Clustering: basics of clustering, k-means clustering, k-means++, methods: elbow. Silhouette coefficient. Dimensionality Reduction: Motivation, PCA, principal components, feature reduction and transformation.

Course Outcomes:

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

CO1: Explain the basic concepts, learning approaches, and mathematical foundations of machine learning.

CO2: Apply data preprocessing, regression models, and regularization techniques to solve prediction problems.

CO3: Build classification models using probabilistic methods, logistic regression, and instance-based learning techniques

CO4: Analyze decision tree, support vector machine, and model evaluation techniques for supervised learning problems

CO5: Apply ensemble learning, clustering, and dimensionality reduction methods for model improvement and pattern discovery.

Text Books:

1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly, 3rd Edition, 2023.
2. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
3. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 4th Edition, 2020.

Reference Books:

1. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong - Mathematics for Machine Learning, 2022
2. Tom M. Mitchell, Machine Learning, McGraw-Hill, 1997.
3. Luis Serrano, Grokking Machine Learning, Manning Publications, 2021.
4. Debugging Machine Learning Models with Python" by Ali Madani (Packt), 2023

Online Learning Resources

1. Mathematics for Machine Learning and Data Science Specialization – https://www.deeplearning.ai/courses/mathematics-for-machine-learning-and-data-science-specialization/
2. Machine Learning Specialization - Andrew Ng (DeepLearning.AI & Stanford) https://www.deeplearning.ai/courses/machine-learning-specialization/
3. NPTEL – Machine Learning (Prof. Balaraman Ravindran, IIT Madras) https://nptel.ac.in/courses/106106139
4. NPTEL – Introduction to Machine Learning (Prof. Sudeshna Sarkar, IIT Kharagpur) https://nptel.ac.in/courses/106105152

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

B. Tech II Year II Semester

25BACAILC04 OBJECT ORIENTED PROGRAMMING USING JAVA LABORATORY

L	T	P	C
0	0	3	1.5

Course Objectives:

The aim of this course is to

1. To provide hands-on experience in developing Java programs using basic programming constructs and object-oriented programming concepts.
2. To understand and implement classes, objects, inheritance, interfaces, abstract classes, constructors, and methods in Java.
3. To develop problem-solving skills using exception handling, packages, string handling, and arrays in Java applications.
4. To understand multithreading concepts, synchronization, daemon threads, and inter-thread communication in Java.
5. To design and implement client-server applications using Java socket programming.

List of Experiments:

1. Write a JAVA program to display default value of all primitive data type of JAVA.
2. Write a Java program using control statements to find the largest of three numbers.
3. Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a Palindrome.
4. Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.
5. Write a JAVA program to implement constructor overloading.
6. Write a JAVA program implement method overloading.
7. Write a JAVA program to implement multi-level Inheritance
8. Write a JAVA program give example for “super” keyword.
9. Write java program to create a super class called Figure that receives the dimensions of two-dimensional objects. It also defines a method called area that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and second is Triangle. Each of the sub class overridden area() so that it returns the area of a rectangle and a triangle respectively.
10. Write a JAVA program for abstract class to find areas of different shapes.
11. Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?
12. Write a JAVA program that import and use the user defined packages
13. Write a JAVA program that describes exception handling mechanism
14. a) Write a JAVA program that creates threads by extending Thread class. First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds, (Repeat the same by implementing Runnable)
    b) Write a program illustrating is Alive and join ()
    c) Write a Program illustrating Daemon Threads.
15. Write a Java program to demonstrate socket communication where a server receives a message from a client and responds back.

Software requirements:

JDK 1.5 or higher versions

Course Outcomes:

After successful completion of the laboratory course, students will be able to:

CO1: Develop Java programs using basic programming constructs, control statements, classes, objects, and methods.

CO2: Apply object-oriented programming concepts such as inheritance, polymorphism, abstract classes, and interfaces in Java applications.

CO3: Implement constructor overloading, method overloading, packages, and exception handling mechanisms in Java programs.

CO4: Design and execute multithreaded Java applications using threads, synchronization, daemon threads, and inter-thread communication.

CO5: Develop Java applications for client-server communication using socket programming.

Reference Books:

1. P. J. Deitel, H. M. Deitel, "Java for Programmers", Pearson Education, PHI, 4th Edition, 2007.
2. P. Radha Krishna, "Object Oriented Programming through Java", Universities Press, 2nd Edition, 2007.
3. Bruce Eckel, "Thinking in Java", Pearson Education, 4th Edition, 2006.
4. Sachin Malhotra, Saurabh Chaudhary, "Programming in Java", Oxford University Press, 5th Edition, 2010.

Online Resources:

1. https://java-iitd.vlabs.ac.in/
2. http://peterindia.net/JavaFiles.html

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech II Year II Semester

25BACAILC05 OPERATING SYSTEMS LABORATORY

L	T	P	C
0	0	3	1.5

Pre-requisite: Nil

Course Objectives:

The objectives of the course are to:

1. To understand the basics of Unix command 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
5. To implement various memory allocation File Allocation methods.

List of Experiments:

1. 0 0 3 1.5 Practicing of Basic UNIX Commands.
2. Write programs using the following UNIX operating system calls fork, exec, getpid, exit, wait, close, stat, opendir and readdir
3. Simulate UNIX commands like cp, ls, grep, etc.,
4. Simulate the following CPU scheduling algorithms a) FCFS b) SJF c) Priority d) Round Robin
5. Control the number of ports opened by the operating system with a) Semaphore b) Monitors.
6. Write a program to illustrate concurrent execution of threads using pthreads library.
7. Write a program to solve producer-consumer problem using Semaphores.
8. Implement the following memory allocation methods for fixed partition a) First fit b) Worst fit c) Best fit
9. Simulate the following page replacement algorithms a) FIFO b) LRU c) LFU
10. Implement Bankers Algorithm for Dead Lock avoidance and prevention
11. Simulate Paging Technique of memory management.
12. Simulate the following file allocation strategies a) Sequential b) Indexed c) Linked Perform Principal Component Analysis on a high-dimensional dataset for dimensionality reduction and visualize the transformed features before and after PCA.

Hardware Requirements: Computer systems with minimum Intel i3 processor, Minimum 4 GB RAM, Minimum 20 GB free hard disk space

Software requirements: Linux OS

Course Outcomes:

After completion of this course, students will be able to:

CO1: Trace different CPU Scheduling algorithms.

CO2: Implement Bankers Algorithms to Avoid the Dead Lock.

CO3: Evaluate Page replacement algorithms.

CO4: Illustrate the file organization techniques.

CO5: Illustrate Inter process Communication and concurrent execution of threads.

Reference Books:

1. 1.“Operating System Concepts”, Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, 10th Edition, John Wiley and Sons Inc., 2018.

Online Resources:

1.“Learn Unix”, https://www.tutorialspoint.com/unix/index.htm
2. “Linux programming and scripting”, https://nptel.ac.in/courses/117/106/117106113/

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech II Year II Semester

25BACAILC06 MACHINE LEARNING LABORATORY

L	T	P	C
0	0	3	1.5

Pre-requisite: Python programming for AI

Course Objectives:

The objectives of the course are to:

1. provide hands-on practice in data loading, preprocessing, exploratory data analysis and feature engineering using benchmark datasets
2. train students to build and evaluate regression and classification models using suitable machine learning tools
3. enable students to compare machine learning models using appropriate performance measures and validation techniques
4. introduce practical implementation of clustering, dimensionality reduction, and ensemble learning methods
5. develop the ability to interpret machine learning results and select suitable models for different datasets

List of Experiments:

1. Implement a program to load any real-world benchmark dataset from a file or URL and perform basic preprocessing operations such as handling missing values, removing duplicates and identifying data types.
2. Perform Exploratory Data Analysis using summary statistics, correlation matrix, histograms, box plots, scatter plots, and heatmaps.
3. Apply feature scaling techniques such as Min-Max normalization, Z-score standardization and compare their effect on numerical attributes.
4. Develop Simple, Multiple Linear Regression models and evaluate their performance using MAE, MSE, RMSE and R^2 score.
5. Construct Polynomial Regression models and compare their performance with linear regression to demonstrate underfitting and overfitting.
6. Build Ridge and Lasso regression models and analyze the effect of regularization strength on model coefficients and prediction error.
7. Develop a Logistic Regression classifier for any binary classification dataset and evaluate the model using confusion matrix, accuracy, precision, recall, F1-score and ROC-AUC.
8. Create a Gaussian Naïve Bayes classifier and compare its performance with Logistic Regression on the same dataset.
9. Apply k-Nearest Neighbor classification using different distance metrics and different values of k.
10. Build Decision Tree model for both classification and regression tasks and examine effects of tree depth, splitting criteria and pruning parameters.
11. Implement Support Vector Machine models using linear, polynomial, and RBF kernels and evaluate their performance using suitable metrics.
12. Perform k-fold cross-validation and hyperparameter tuning using GridSearchCV or RandomizedSearchCV for any supervised learning model.

13. Build and compare ensemble learning models such as Random Forest, AdaBoost, and Gradient Boosting using a suitable dataset.
14. Conduct clustering analysis using k-Means and k-Means++ and determine the suitable number of clusters using elbow method
15. Perform Principal Component Analysis on a high-dimensional dataset for dimensionality reduction and visualize the transformed features before and after PCA.

Hardware Requirements: Computer systems with minimum Intel i3 processor, Minimum 4 GB RAM, Minimum 20 GB free hard disk space

Software requirements: Python 3.x, Jupyter Notebook / Google Colab, NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, SciPy, OpenML / Kaggle / UCI ML Repository datasets, Windows / Linux / macOS operating system

Course Outcomes:

After completion of this course, students will be able to:

CO1: Perform data loading, cleaning, preprocessing, and exploratory data analysis using Python libraries.
CO2: Build and evaluate regression models such as linear regression, polynomial regression, Ridge regression, and Lasso regression.
CO3: Develop classification models such as Logistic Regression, Naïve Bayes, k-NN, Decision Tree, and Support Vector Machine.
CO4: Compare machine learning models using suitable evaluation metrics, cross-validation, and hyperparameter tuning methods.
CO5: Apply clustering, dimensionality reduction, and ensemble learning techniques for data analysis and model improvement.

Reference Books:

1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly, 3rd Edition, 2023.
2. Debugging Machine Learning Models with Python" by Ali Madani (Packt), 2023
3. M. N. Murty and Ananthanarayana V. S., Machine Learning: Theory and Practice, University Press.
4. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 4th Edition, 2020.

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

Skill Enhancement Courses

Skill Enhancement Course - I

25BACAISC01 FULL STACK DEVELOPMENT - I

L	T	P	C
1	0	2	2

Course Objectives:

1. Make use of HTML elements and their attributes for designing static web pages
2. Build a web page by applying appropriate CSS styles to HTML elements
3. Experiment with JavaScript to develop dynamic web pages and validate forms
4. Build a web page using UI/UX design

UNIT I — 6 hours

HTML Basics - Lists - Links – Images - HTML Tables - Forms – Frames

a. Design a webpage for a restaurant. The webpage should include the following elements:
   - An ordered list to describe the steps to make a reservation at the restaurant.
   - An unordered list to display the main categories of the menu (e.g., Appetizers, Main Courses, Desserts).
   - Nested lists to show sub-categories within each main category (e.g., under Main Courses, list "Vegetarian", "Non-Vegetarian", and "Vegan").
   - An ordered list within an unordered list to detail a special meal plan (e.g., daily special menu with steps for preparation).
   - Include links to the restaurant's social media pages (e.g., Facebook, Instagram) using the <a> tag with the href attribute.
   - Use the target attribute to open these links in a new tab.
   - Display an image of the restaurant's special dish with specified dimensions (height and width) and make the image a clickable link that navigates to a page with more details about the dish.

b. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)

c. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).

d. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using <select>&<option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).

e. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame image, second frame paragraph, third frame hyperlink. And also make sure of using "no frame" attribute such that frames to be fixed).

UNIT II — 6 hours

HTML 5 - Cascading Style Sheets, Types of CSS - Selector forms -5. CSS with Color, Background, Font, Text and CSS Box Model

a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, , tags.

b. Write a HTML program, to embed audio and video into HTML web page.

c. Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

d. Write a program to apply different types of selector forms
   - Simple selector (element, id, class, group, universal)
   - Combinator selector (descendant, child, adjacent sibling, general sibling)
   - Pseudo-class selector
   - Pseudo-element selector

- Attribute selector

e. Write a program to demonstrate the various ways you can reference a color in CSS.
f. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.
g. Write a program using the following terms related to CSS font and text:
   i. font-size ii. font-weight iii. font-style
   iv. text-decoration v. text-transformation vi. text-alignment
h. Write a program, to explain the importance of CSS Box model using
   i. Content ii. Border iii. Margin iv. padding

UNIT III **6 hours**

Introduction to Javascript - Applying JavaScript - internal and external, I/O, Type Conversion - JavaScript Pre-defined and User-defined Objects

a. Write a program to embed internal and external JavaScript in a web page.
b. Write a program to explain the different ways for displaying output.
c. Write a program to explain the different ways for taking input.
d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not
e. Write a program using the following object properties and methods.
   i. Document object
   ii. array object
   iii. math object
   iv. string object
   v. regex object
   vi. date object
f. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

UNIT IV **6 hours**

JavaScript Conditional Statements and Loops - Javascript Functions and Events

a. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words “LARGER NUMBER” in an information message dialog. If the numbers are equal, output HTML text as “EQUAL NUMBERS”.
b. Write a program to display week days using switch case.
c. Write a program to print 1 to 10 numbers using for, while and do-while loops.
d. Write aprogram to print data in object using for-in, for-each and for-of loops
e. Develop a program to determine whether a given number is an ‘ARMSTRONG NUMBER’ or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]
f. Write a program to display the denomination of the amount deposited in the bank in terms of 100’s, 50’s, 20’s, 10’s, 5’s, 2’s & 1’s. (Eg: If deposited amount is Rs.163, the output should be 1-100’s, 1-50’s, 1- 10’s, 1-2’s & 1-1’s)
g. Design a HTML having a text box and four buttons named Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate function should be called to display
   1. Factorial of that number
   2. Fibonacci series up to that number
   3. Prime numbers up to that number
   4. Is it palindrome or not
h. Write a program to validate the following fields in a registration page
   i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters)
   ii. Mobile (only numbers and length 10 digits)

iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

UNIT V **6 hours**

UI/UX Design Principles and Techniques

a. Explain the basic principles of UI/UX design including user research, wireframing, prototyping, and usability testing.
b. Create a wireframe for a simple website layout using a tool like Balsamiq or Figma.
c. Develop a prototype of the website layout created in the wireframe using HTML and CSS.

Course Outcomes:

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

CO1: Proficient in HTML and CSS for Static Web Pages
CO2: Build dynamic Web Page Development with JavaScript
CO3: Implement effective UI/UX Design

Text Books:

1. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.
2. Web Programming with HTML5, CSS and JavaScript, John Dean, Jones & Bartlett Learning, 2019 (Chapters 1-11).

Reference Books:

1. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.

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

Skill Enhancement Course - II

25BACAISC02 FULL STACK DEVELOPMENT - II

L	T	P	C
1	0	2	2

Pre-requisite: NIL

Course Objectives:

1. To become knowledgeable about the most recent web development technologies.
2. Idea for creating two tier and three tier architectural web applications.
3. Design and Analyse real time web applications.
4. Constructing suitable client and server side applications.
5. To learn core concept of both front end and back end programming.

UNIT I WEB DEVELOPMENT BASICS 6 hours

Web development Basics - HTML & Web servers Shell - UNIX CLI Version control - Git & Github HTML, CSS.

a. Develop a personal portfolio webpage using basic HTML and CSS
b. Initialize a Git repository and push a static web project to GitHub
c. Practice basic Linux/Unix commands using terminal (navigation, file operations)
d. Create a multi-page static website with navigation and layout
e. Demonstrate version control operations: commit, branch, merge, clone

UNIT II FRONTEND DEVELOPMENT 6 hours

Javascript basics OOPS Aspects of JavaScript Memory usage and Functions in JS AJAX for data exchange with server jQuery Framework jQuery events, UI components etc. JSON data format.

a. Write JavaScript programs demonstrating variables, functions, and objects
b. Develop an AJAX-based form to submit and retrieve data from a mock server
c. Build a dynamic form with client-side validation using jQuery
d. Design interactive UI elements using jQuery (sliders, modals, tabs)
e. Work with JSON data for dynamic rendering of content

UNIT III REACT JS 6 hours

Introduction to React React Router and Single Page Applications React Forms, Flow Architecture and Introduction to Redux More Redux and Client-Server Communication.

a. Set up a basic React project using Create React App
b. Implement routing between pages using React Router
c. Create and manage form data using React state and hooks
d. Use Redux to manage state in a to-do or counter app
e. Fetch and render data from a public API using Axios

UNIT IV JAVA WEB DEVELOPMENT 6 hours

JAVA programming basics, Model View Controller (MVC) Pattern MVC Architecture using Spring RESTful API using Spring Framework Building an application using Maven.

a. Create a Java console app demonstrating classes, objects, and methods
b. Build a simple web application using Spring MVC pattern

c. Develop RESTful web services using Spring Boot (CRUD operations)
d. Connect frontend to Spring backend with JSON-based communication
e. Use Maven to build, package, and run the Spring Boot application

UNIT V DATABASES & DEPLOYMENT 6 hours

Relational schemas and normalization Structured Query Language (SQL) Data persistence using Spring JDBC Agile development principles and deploying application in Cloud.
a. Design and normalize a database schema up to 3NF
b. Perform SQL-based CRUD operations on MySQL/PostgreSQL
c. Integrate MySQL with a Spring Boot application using Spring JDBC
d. Plan and manage tasks using Agile tools (Trello, Jira)
e. Deploy a full stack application on a cloud platform (Heroku, AWS, or GitHub Pages)

Course Outcomes:

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

CO1: Demonstrate about the most recent web development technologies.

CO2: Develop two tier and three tier architectural web applications.

CO3: Implement real time web applications.

CO4: Apply suitable client and server-side applications.

CO5: Develop core concept of both front end and back-end programming.

Text Books:

1. Web Design with HTML, CSS, JavaScript and JQuery Set Book by Jon Duckett Professional JavaScript for Web Developers Book by Nicholas C. Zakas
2. "Professional JavaScript for Web Developers", Zakas N C, Wiley, 3rd Edition, 2011.
3. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites by Robin Nixon

Reference Books:

1. Full-Stack JavaScript Development by Eric Bush
2. Mastering Full Stack React Web Development Paperback – April 28, 2017 by Tomasz Dyl, Kamil Przeorski, Maciej Czarnecki

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