MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(Deemed to be University)

MADANAPALLE

www.mits.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

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 (DATA SCIENCE)

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(Deemed to be University)

MADANAPALLE

B. Tech Four Year Curriculum Structure

Branch: COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

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	25BACSELC02	IT Essentials Laboratory	0	0	2	2	1
Total				**16**	**0**	**8**	**24**	**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	25BACDSTC01	Fundamentals 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	25BACOMLC01	Engineering Skills Laboratory	0	0	2	2	1
10	ESC	25BACSELC03	Data Structures and Algorithms Laboratory	0	0	2	2	1
Total				**16**	**0**	**8**	**24**	**20**

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

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	25BACDSTC02	Database Systems	3	0	0	3	3
4	PCC	25BACDSTC03	Object-Oriented Design and Programming	3	0	0	3	3
5	PCC	25BACDSTC04	Algorithm Design and Analysis	3	0	0	3	3
6	PCC	25BACDSLC01	Database Systems Laboratory	0	0	2	2	1
7	PCC	25BACDSLC02	Object-Oriented Design and Programming Laboratory	0	0	2	2	1
8	PCC	25BACDSLC03	Algorithm Design and Analysis 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 – I** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**16**	**0**	**8**	**24**	**20**

II Year II Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	BSC	25BAMATTC12	Discrete Mathematical Structures	3	0	0	3	3
2	PCC	25BACDSTC05	Data Engineering	3	0	0	3	3
3	PCC	25BACDSTC06	Applied Machine Learning	3	0	0	3	3
4	PCC	25BACDSTC07	Operating Systems Fundamentals	3	0	0	3	3
5	OE		**Open Elective – I** *(Refer ANNEXURE - II)*	3	0	0	3	3
6	PCC	25BACDSLC04	Data Engineering Laboratory	0	0	2	2	1
7	PCC	25BACDSLC05	Applied Machine Learning Laboratory	0	0	2	2	1
8	PCC	25BACDSLC06	Operating Systems Fundamentals 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 – II** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**16**	**0**	**10**	**26**	**20**

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

R25 – Tentative Curriculum Structure
III Year I Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	HSMC		Entrepreneurship and Start-ups	3	0	0	3	3
2	PCC	25BACDSTC08	Big Data Analytics	3	0	0	3	3
3	PCC	25BACDSTC09	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	25BACDSLC07	Big Data Analytics Laboratory	0	0	2	2	1
7	PCC	25BACDSLC08	Deep Learning Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – III** *(Refer ANNEXURE - IV)*	1	0	2	3	2
9	INT	25BACDSIC01	Summer Internship - I	0	0	2	2	1
10	EEC		**NSDC - Certification Course – III** *(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		Financial Technology (FinTech)	3	0	0	3	3
2	PCC	25BACDSTC10	Data Mining and Data Warehousing	2	0	0	2	2
3	PCC	25BACDSTC11	Theory of Computation	3	0	0	3	3
4	PCC	25BACDSTC12	Data Visualization	2	0	0	2	2
5	DE		**Discipline Elective – II** *(Refer ANNEXURE - III)*	3	0	0	3	3
6	OE		**Open Elective – III** *(Refer ANNEXURE - II)*	3	0	0	3	3
7	PCC	25BACDSLC09	Data Mining and Data Warehousing Laboratory	0	0	2	2	1
8	PCC	25BACDSLC10	Data Visualization Laboratory	0	0	2	2	1
9	AEC	25BAENGEC01	Soft Skills (Career Essentials)	1	0	2	3	2
10	EEC		**NSDC - Certification Course – IV** *(Refer ANNEXURE - V)*	0	0	2	2	-
Total				**17**	**0**	**8**	**25**	**20**

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

R25 - Tentative Curriculum Structure
IV Year I Semester

S. No.	Category	Course Code	Course Title	Hours Per Week				Credits
				L	T	P	Total	
1	PCC	25BACDSTC13	IoT and Cloud Computing	3	0	0	3	3
2	PCC	25BACDSTC14	Computer Networks and Internet Protocols	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	25BACDSLC11	IoT and Cloud Computing Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – IV** *(Refer ANNEXURE – IV)*	1	0	2	3	2
9	INT	25BACDSIC02	Summer Internship - II	0	0	4	4	2
10	EEC		**NSDC - Certification Course – V** *(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	25BACDSPC01	Capstone Project	0	0	28	28	14
2	INT	25BACDSIC03	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.	25BACDSDM01	Software Engineering
2.	25BACDSDM02	Natural Language Processing
3.	25BACDSDM03	Cyber Security and Privacy
4.	25BACDSDM04	Scalable Data Science
Any advanced Course offered by SWAYAM NPTEL can be appended in future.		

Discipline Elective – II		
S. No.	**Course Code**	**Course Title**
1.	25BACDSDC01	Software Testing
2.	25BACDSDC02	Introduction to Large Language Models
3.	25BACDSDC03	Cryptography and Network Security
4.	25BACDSDC04	NoSQL Databases
Any advanced courses can be appended in future.		

Discipline Elective – III		
S. No.	**Course Code**	**Course Title**
1.	25BACDSDC05	Software Project Management
2.	25BACDSDC06	Reinforcement Learning
3.	25BACDSDC07	Blockchain Technology
4.	25BACDSDC08	Social Network Analysis
Any advanced courses can be appended in future.		

Discipline Elective – IV		
S. No.	**Course Code**	**Course Title**
1.	25BACDSDC09	Agile Methodologies
2.	25BACDSDC10	Expert Systems
3.	25BACDSDC11	Recommender Systems
4.	25BACDSDC12	Responsible Artificial Intelligence
Any advanced courses can be appended in future.		

Discipline Elective –V		
S. No.	**Course Code**	**Course Title**
1	25BACDSDC13	Software Architecture and Design Pattern
2	25BACDSDC14	Optimization Techniques for AI
3	25BACDSDC15	High Performance Computing
4	25BACDSDC16	Explainable Artificial Intelligence (XAI)
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.	25BACDSSC01	Python for Data Science
Any Courses can be appended in future.		

Skill Enhancement Course – II		
S. No.	**Course Code**	**Course Title**
1.	25BACDSSC02	Full Stack Development
Any Courses can be appended in future.		

Skill Enhancement course – III		
Sl. No.	**Course Code**	**Course Title**
1.	25BACDSSC03	Devops
Any Courses can be appended in future.		

Skill Enhancement Course – IV		
Sl. No.	**Course Code**	**Course Title**
1.	25BACDSSC04	Generative AI/MLOps
Any Courses can be appended in future.		

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

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

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

25BACDSTC01 FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE

L	T	P	C
3	0	0	3

Pre-requisite NIL

Course Description:

This course introduces the fundamentals of Artificial Intelligence, including its history, data foundations, and basics of Machine Learning. It explores real-world applications in healthcare, finance, and communication, along with ethical issues, responsible AI, and emerging trends for the future.

Course Objectives:

This course enables students to

1. Understand the fundamentals, history, and evolution of Artificial Intelligence.
2. Recognize the role of data and basic statistics in intelligent systems.
3. Explain the basic concepts, types, and tools of Machine Learning.
4. Explore real-world applications of AI/ML in domains such as healthcare, finance, and communication.
5. Discuss ethical implications and emerging trends of AI in society.

UNIT I INTRODUCTION TO AI 9 hours

History & Evolution of AI, Types of AI-Narrow, General, Super, AI in Everyday Life -chatbots, assistants, recommendation systems, Branches of AI-Problem Solving, Reasoning, Robotics, Language, Ethical & Societal Implications-jobs, bias, privacy, security.

Case Study: Deep Blue vs Kasparov (Chess), Self-Driving Cars

UNIT II FOUNDATIONS OF INTELLIGENT SYSTEMS 9 hours

Role of Data in AI systems, Data vs Information vs Knowledge, Types of Data - Structured, Semi-structured, Unstructured, Basics of Statistics - Mean, Median, Mode, Variance, Importance of Data Quality.

Case Study: Weather Prediction with data.

UNIT III BASICS OF MACHINE LEARNING 9 hours

What is ML? Relation between AI, ML, and Data Science. Types of ML-Supervised, Unsupervised, Reinforcement, Learning Cycle-Training, Testing, Feedback, Introduction to ML Tools (Weka, Teachable Machine, Orange).

Case Study: Email Spam Filtering, Netflix Recommendations.

UNIT IV AI & ML IN THE REAL WORLD 9 hours

Applications of AI/ML- image recognition, NLP (Chatbots, Translation), recommendation systems, Finance (Fraud Detection, Credit Scoring, Stock Market Prediction), Predictive Analytics, Challenges in AI- Bias, Fairness, Privacy, Data Quality, Responsible AI-Trust, Transparency, and Job Automation.

Case Study: Healthcare diagnosis using ML, AI in Banking Fraud Detection.

UNIT V EMERGING TRENDS IN AI 9 hours

Future of AI: General AI, Super AI (conceptual only), AI in Emerging Applications: Smart Homes, Smart Cities, Healthcare, Education, AI for Social Good: agriculture, disaster prediction, accessibility, Limitations of AI today.

Case Study: AI in Climate Change, AI in Education.

Course Outcomes:
By the end of the course, students will be able to
CO1: Describe the basics, history, and types of Artificial Intelligence.
CO2: Interpret data concepts and apply simple statistical measures.
CO3: Explain the fundamental types of Machine Learning and the ML process.
CO4: Identify and analyse real-world applications of AI/ML.
CO5: Evaluate emerging trends, applications for social good, and limitations of AI.

Text Books

1. Russell, Stuart Jonathan, Norvig, Peter, Davis, Ernest. Artificial Intelligence: A Modern Approach. United Kingdom: Pearson, 2010.
2. Deepak Khemani. A First Course in Artificial Intelligence. McGraw Hill Education (India), 2013.

Reference Books

1. Practical Statistics for Data Scientists, 2nd Edition by Peter Bruce,Andrew Bruce, Peter Gedeck

2. Denis Rothman. Artificial Intelligence by Example, Packt, 2018.

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

B. Tech I Year II Semester

25BAENGLC01 PROFESSIONAL COMMUNICATION LABORATORY

L	T	P	C
0	0	2	1

Course Description:

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

Course Objectives:

This course aims to:

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

Lab Activities

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

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

Course Outcomes:

Students will be able to:

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

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

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

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

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

Text Books:

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

Reference Books:

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

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech I Year II Semester

25BAPHYLC01 PHYSICS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisites: Plus two physics knowledge

Course Description:

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

Course Objectives:

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

List of Experiments:

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

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

Course Outcomes:

CO1: Know the various phenomena of light practically and gain knowledge about various optical technique methods.

CO2: Verify the theoretical concepts of optics, magnetism, waves and oscillations by hands on experiment.

CO3: Apply the scientific process in the conducting of experiments and report the experimental findings.

CO4: Understand mechanical phenomena by instruments and apply them in real time applications.

CO5: Acquire and interpret experimental data to examine the physical laws.

Text / Reference Books:

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

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

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech I Year II Semester

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.

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.

II Year I Semester

25BACOMTC01 COMPUTER ORGANIZATION AND ARCHITECTURE

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Description:

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

Course Objectives:

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

UNIT I OVERVIEW & INSTRUCTIONS 9 hours

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

UNIT II ARITHMETIC OPERATIONS 9 hours

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

UNIT III PROCESSOR AND CONTROL UNIT 9 hours

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

UNIT IV PARALLELISM 9 hours

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

UNIT V MEMORY AND I/O SYSTEMS 9 hours

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

Course Outcomes:

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

Text Books:

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

Reference Books:

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

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

B.Tech II Year I Semester

25BACDSTC02 DATABASE SYSTEMS

L	T	P	C
3	0	0	3

Pre-requisite: None

Course Description:
This course introduces advanced concepts and practices in database management systems. It covers relational, SQL, NoSQL, and cloud-based database systems, emphasizing data modeling, indexing, transactions, concurrency control, and database security. Students learn query optimization with Web frameworks. Practical sessions focus on SQL, NoSQL databases, and database administration tools used in contemporary applications.

Course Objectives:
This course enables students

1. To learn the fundamentals of data models, relational algebra and SQL
2. To represent a database system using ER diagrams and to learn normalization techniques
3. To understand the fundamental concepts of transaction, concurrency and recovery processing
4. To understand the internal storage structures using different file and indexing techniques
5. To have an introductory knowledge about the Distributed databases and NOSQL database

UNIT I RELATIONAL DATABASES 9 hours

Overview of RDBMS – Purpose of Database System – Views of data – Data Models – Database System Architecture – Introduction to relational databases – Relational Model – Keys – Relational Algebra – SQL fundamentals – Advanced SQL features – Embedded SQL– Dynamic SQL

UNIT II DATABASE DESIGN 9 hours

Entity-Relationship model – E-R Diagrams – Enhanced-ER Model – ER-to-Relational Mapping – Functional Dependencies – Non-loss Decomposition – First, Second, Third Normal Forms, Dependency Preservation – Boyce/Codd Normal Form – Multi-valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form

UNIT III TRANSACTION PROCESSING 9 hours

Transaction Concepts – ACID Properties – Schedules – Serializability – Transaction support in SQL – Need for Concurrency – Concurrency control –Two Phase Locking- Timestamp – Multiversion – Validation and Snapshot isolation– Multiple Granularity locking – Deadlock Handling – Recovery Concepts – Recovery based on deferred and immediate update – Shadow paging – ARIES Algorithm

UNIT IV IMPLEMENTATION TECHNIQUES 9 hours

RAID – File Organization – Organization of Records in Files – Data dictionary Storage – Column Oriented Storage– Indexing and Hashing –Ordered Indices – B+ tree Index Files – B tree Index Files – Static Hashing – Dynamic Hashing – Query Processing Overview – Algorithms for Selection, Sorting and join operations – Query optimization using Heuristics - Cost Estimation.

UNIT V ADVANCED APPLICATION DEVELOPMENT AND RECENT TREND **9 hours**

Advanced Application Development – Connecting to MongoDB with Python, MongoDB query Language, Updating/Deleting documents in collection, MongoDB query operators. Distributed Databases: Architecture– NOSQL Databases: Introduction – CAP Theorem – Document Based systems – Key value Stores – Column Based Systems – Graph Databases. Database Security: Security issues – Access control based on privileges – Role Based access control – SQL Injection – Statistical Database security – Flow control – Encryption and Public Key infrastructures – Challenges

Course Outcomes:

After completing this Unit, students will be able to

CO1: Construct SQL Queries using relational algebra
CO2: Design database using ER model and normalize the database
CO3: Construct queries to handle transaction processing and maintain consistency of the database
CO4: Compare and contrast various indexing strategies and apply the knowledge.
CO5: Appraise how advanced databases differ from Relational Databases and find a suitable database for the given requirement.

Text Book(s)

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, "Database System Concepts", Seventh Edition, McGraw Hill, 2020.
2. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Seventh Edition, Pearson Education, 2017
3. Andreas Meier, Michael Kaufmann, "SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management", Springer Verlag 2019

Reference Books

1. C.J.Date, A.Kannan, S.Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.
2. Raghu Ramakrishnan and Johannas Gehrke, "Database Management Systems", Third Edition, McGraw-Hill, 2003.

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

B. Tech II Year I Semester

25BACDSTC03 OBJECT-ORIENTED DESIGN AND PROGRAMMING

L	T	P	C
3	0	0	3

Pre-requisite: Course on programming for problem solving.

Course Description:

This course introduces the principles of Object-Oriented Programming using Java, focusing on classes, objects, inheritance, polymorphism, abstraction, and encapsulation. Learners develop robust, modular, and reusable applications through hands-on coding with core Java APIs and exception handling.

Course Objectives:

This course enables students to

1. To understand and apply core object-oriented principles such as abstraction, encapsulation, inheritance, and polymorphism using Java.
2. To design and implement Java programs using classes, objects, constructors, and methods.
3. To develop robust applications by applying exception handling, multithreading, and file I/O concepts.
4. Enable students to understand and implement Java exception handling mechanisms
5. Develop their ability to design and implement multithreaded Java applications

UNIT I INTRODUCTION TO OBJECT ORIENTED PROGRAMMING 9 hours

Object Oriented Programming Concepts, Compare with procedure-oriented programming, Features of Java Language, Types of Java Programs, Java Architecture (JDK, JRE and JVM), Structure of Java Program, Literals, Data Types, Variables, Operators, if else statement, switch statement, while, do while, break, continue.

UNIT II ARRAYS AND CLASSES 9 hours

One-Dimensional Array, Multi-Dimensional Array, String Class and Methods, StringBuffer Class and methods, Defining a Class, the new Operator, Objects, the dot operator, Method Declaration and Calling, Instance Variable, Local Variables, Constructors, Types of constructors, Constructor overloading, this keyword, Method Overloading, Passing Objects as Parameters,

UNIT III INHERITANCE AND POLYMORPHISM 9 hours

Creating Subclasses, Method Overriding, Dynamic Method Dispatch, super Keyword, final keyword, Static Class, Methods, Block and Variables, Abstract Class and method, Interfaces- Creating, Implementing, Using, Extending, and Nesting of interfaces.

UNIT IV PACKAGES & EXCEPTION HANDLING 9 hours

Packages: Defining, Finding and Importing packages, Member Access. Exception Handling: Fundamentals, Types, Multiple catch clauses, Nested try blocks, Thrown Class, Using Finally and Throws, Built-in exceptions, User-defined exceptions. I/O Streams: Byte Stream Classes and Character Stream Classes.

UNIT V MULTITHREADED PROGRAMMING 9 hours

Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Interthread Communication - Suspending, Resuming, and Stopping of Threads. Applet- Introduction, Life Cycle of an Applet, GUI with an Applet, Abstract Window Toolkit (AWT)

Course Outcomes:

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

CO1: Analyze problems, design solutions using OOP principles, and implement them efficiently in Java.

CO2: Design and implement classes to model real-world entities, with a focus on attributes, behaviours, and relationships between objects.

CO3: Demonstrate an understanding of inheritance hierarchies and polymorphic behaviour, including method overriding and interface concept.

CO4: Apply Competence in handling exceptions and errors to write robust and fault-tolerant code.

CO5: Develop multithreaded applications with synchronization.

Text Book(s)

1. Java - The Complete Reference By **Herbert Schildt and Danny Coward**, 13th edition, McGraw-Hill in **2024**.
2. **Object-Oriented Programming Using Java** — Tripti Paul, 1st Edition,2018, Cengage Learning India Pvt. Ltd.

Reference Books

1. Programming with Java – A primer by E. Balagurusamy, 7th Edition, Tata Mc-Graw Hill Publication,2023
2. Object-Oriented Programming Using Java- Tripti Paul, Cengage publication, 1st Edition, 2018
3. Java How to Program- Paul Deitel & Harvey Deitel, Pearson / Prentice Hall ,11th Edition ~2018
4. Core Java Volume-I Fundamentals by Horstmann & Cornell, Eight Edition, Pearson Education,2019

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

B. Tech II Year I Semester

25BACDSTC04 ALGORITHM DESIGN AND ANALYSIS

L	T	P	C
3	0	0	3

Pre-requisite 25BACSETC03

Course Description:

This course provides a foundational understanding of algorithmic principles and their applications, emphasizing problem-solving and efficiency analysis. Students will explore algorithm design techniques, including Divide and Conquer, Greedy, and Dynamic Programming, gaining practical skills in solving complex computational problems. The course delves into state space searching methods, fostering proficiency in graph algorithms, backtracking, and branch and bound. The exploration of NP-Hard, NP-Complete problems, and approximation algorithms equips students to tackle inherently complex computational challenges.

Course Objectives:

This course enables students to

1. Impart foundational understanding of algorithmic problem-solving, including the basics of algorithm design, analysis.
2. Introduce two key algorithmic design paradigms—Divide and Conquer and Greedy method
3. Create a comprehensive understanding of dynamic programming principles and apply them to solve optimization problems.
4. Teach the skills in state space searching through graph search algorithms, backtracking, and branch and bound methods.
5. Familiarize the student with the complexities of NP-hard and NP-complete problems.

UNIT I FOUNDATIONS OF ALGORITHM ANALYSIS AND NON-LINEAR DATA STRUCTURES **9 hours**

Introduction to Algorithm Analysis, Space and Time Complexity Analysis, Asymptotic Notations. AVL Trees – Creation, Insertion, Deletion Operations and Applications, B-Trees – Creation, Insertion,Deletion operations and Applications. Red Black trees - Properties of Red-Black Trees – Rotations – Insertion – Deletion. Heap – Heap Implementation

UNIT II DIVIDE AND CONQUER & GREEDY METHOD **9 hours**

Recurrences - The Substitution Method – The Recursion Tree Method - Masters Theorem Method. Divide and Conquer: The General Method, Quick Sort, Merge Sort, Strassen's matrix multiplication. Greedy Method: General Method, Job Sequencing with deadlines, Knapsack Problem, Single-Source Shortest paths – Dijkstra's Algorithm, Minimum cost spanning trees

UNIT III DYNAMIC PROGRAMMING **9 hours**

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

UNIT IV BACK TRACKING & BRANCH AND BOUND **9 hours**

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

UNIT V UNRAVELING NP-HARD, NP-COMPLETE PROBLEMS 9 hours

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

Problems: Scheduling Identical Processors, Job Shop Scheduling

Course Outcomes:

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

CO1: Understand algorithm analysis principles and illustrate the operations of advanced tree and graph data structures with relevant applications.

CO2: Design Divide and Conquer strategy and Greedy method for different problems.

CO3: Demonstrate Dynamic programming method to solve problems.

CO4: Apply Backtracking and Branch & bound strategy to solve problems.

CO5: Understand NP-Hard and NP-Complete problem

Text Book(s)

1. "Fundamentals of Computer Algorithms", Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Universities Press, 2nd Edition, 2008.
2. "Introduction to the Design and Analysis of Algorithms", Anany Levitin, Pearson Education, 3rd Edition, 2012

Reference Books

1. "Fundamentals of Data Structures in C++", Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, Universities Press, 2nd Edition, 2008
2. "Introduction to Algorithms", Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Prentice Hall of India, 3rd Edition, 2009.
3. "Data Structures and Algorithms", Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Pearson Education, Reprint Edition, 2006.
4. "The Algorithm Design Manual", Steven S. Skiena, Springer, 3rd Edition, 2020.

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

B. Tech II Year I Semester

25BACDSLC01 DATABASE SYSTEMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Nil

Course Description:

This Database Management Systems laboratory provides hands-on experience with database concepts, including relational database design, query optimization, transaction management, and recovery techniques. Students work with modern database technologies such as Object-Relational DBMS, NoSQL databases (MongoDB, CouchDB), and explore indexing, tuning, and advanced SQL features. The course also covers application development using Python with MongoDB and web framework integration, preparing students for research and industry-oriented database solutions.

Course Objectives:

1. To provide hands-on experience with RDBMS, advanced SQL, and database tuning techniques.
2. To practice transaction management, concurrency control, and recovery methods.
3. To explore Object-Relational and NoSQL databases with real-world applications.
4. To develop applications integrating databases with modern programming languages like Python.
5. To learn about the various tools available such as MongoDB, Cassandra and perform CRUD operations

List of Experiments:

1. **Installation and Configuration of RDBMS**
   Install MySQL/PostgreSQL/Oracle and configure client-server environment
2. **Table Creation and Constraints**
   Create a database table, add constraints (primary key, unique, check, Not null), insert rows, update and delete rows using SQL DDL and DML commands.
3. **Relational Algebra and Relational Calculus**
   i. Write SQL equivalents for relational algebra operations (SELECT, PROJECT, JOIN, etc.).
   ii. Practice tuple and domain relational calculus queries.
4. **Join operations**
   Query the database tables and explore sub queries and simple join operations, natural, equi and outer joins.
5. **Trigger**
   Write SQL Triggers for insert, delete, and update operations in a database table.
6. **Functions and Procedure**
   Write user defined functions and stored procedures in SQL.
7. **Views**
   Create View and index for database tables with a large number of records.
8. **NoSQL Database Operations**
   i. Installation and setup of MongoDB and CouchDB.
   ii. Create collections, perform CRUD operations, and explore CAP Theorem with query models.
9. **Document Creation**
   Create Document, column and graph based data using NOSQL database tools
10. **MongoDB with Python Integration**

i. Connect MongoDB with Python using PyMongo.
ii. Perform insert, update, delete, and retrieve operations.

11. Indexing and Advanced Query Operators in MongoDB
Implement GeoSpatial Indexing, Upserts, and query operators ($gt, $lt, $in, $regex, etc.).

12. Web Framework Integration
Develop a simple web application using Python Flask/Django connected to MongoDB.

Course Outcomes:

Upon successful completion, students will be able to:

CO1: Design normalized relational schemas and implement them using SQL.
CO2: Optimize queries and perform database tuning for performance improvement.
CO3: Implement transaction processing and concurrency control mechanisms.
CO4: Work with Object-Relational and NoSQL database models.
CO5: Develop applications integrating MongoDB and Python for advanced use cases.

Text Books:

1. Silberschatz A, Korth H F and Sudharshan S, "Database System Concepts", Sixth Edition, Tata McGraw-Hill Publishing Company Limited, 2010.
2. Andreas Meier, Michael Kaufmann, "SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management", Springer Verlag 2019
3. Ramez Elmasri & Shamkant B. Navathe – *Fundamentals of Database Systems* 3rd Edition, Addison-Wesley,Year: 2000 .

References Books:

1. Raghu Ramakrishnan and Johannas Gehrke, "Database Management Systems", Third Edition, McGraw-Hill, 2003.
2. Hector Garcia-Molina, Jeff Ullman and Jennifer Widom, "Database Systems: The Complete Book", Pearson, 2011

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

B. Tech II Year I Semester

25BACDSLC02 OBJECT-ORIENTED DESIGN AND PROGRAMMING LABORATORY

L	T	P	C
0	0	2	1

Pre Requisites: Basic knowledge of programming concepts using C or Python.

Course Description:

This course introduces the fundamentals of Object-Oriented Programming using Java. It covers core concepts such as classes, objects, inheritance, polymorphism, and abstraction. Students learn to design modular, reusable, and robust programs using Java. The course also emphasizes exception handling and real-world application development.

Course Objectives:

This course enables students to

1. Practice object-oriented programming concepts using the Java programming language.
2. Design and implement classes, objects, methods, inheritance, and interfaces for real-world applications.
3. Apply packages and exception handling mechanisms to develop modular and robust Java programs.
4. Construct and manage multithreaded applications and understand the life cycle and states of threads.
5. Develop GUI-based applications using Applets and Abstract Window Toolkit (AWT) components.

List of Experiments:

1. A banking application accepts a transaction amount entered by a user. The system validates the input and determines whether the amount is positive, negative, or zero, and displays the result.
2. A recruitment system evaluates the test scores of three candidates. The system compares the scores and identifies the highest score among them.
3. An e-learning platform provides practice exercises for students. When a user enters a number, the system generates and displays the multiplication table for that number.
4. An organization maintains employee salary details in an array structure. The system analyzes the data to find and display the maximum and minimum salary values.
5. A library management system stores information such as book ID, title, and price. Each book is treated as an object, and the system displays the stored book details using appropriate constructors.
6. A text-processing application performs various operations on user-input strings. The system calculates string length, compares strings, and modifies text using suitable string manipulation techniques.
7. A transport management system handles different vehicle types such as cars and bikes. All vehicles share common properties, but their behaviors differ based on the type.
8. A digital payment system supports multiple payment modes, including credit card and UPI. The selected payment method is executed dynamically during runtime.
9. A college admission system verifies student age during registration. If the entered age is invalid, the system raises and handles appropriate custom exceptions.
10. A text editor application reads data from files specified by the user. The system ensures smooth execution by handling possible file-related runtime errors.

11. A software application organizes utility classes into reusable modules. This approach enhances modularity, code reuse, and access control through structured packaging.
12. An online booking system processes multiple user requests at the same time. Each booking operation runs independently to ensure efficient and concurrent execution.
13. A simple calculator interface is required for a web-based utility tool. The interface should be developed using an applet with AWT components to accept user input and perform basic arithmetic operations through GUI controls.

Software tools: JDK 8 or above (Recommended: JDK 11 / JDK 17)

Course Outcomes:

After completion of the course, students will be able to

CO1: Demonstrate understanding of Java syntax, data types, control structures, classes, objects, inheritance, polymorphism, and exception handling.
CO2: Apply object-oriented programming principles such as encapsulation, inheritance, abstraction, and polymorphism to solve real-world problems.
CO3: Develop modular Java applications using packages and handle runtime errors using appropriate exception handling mechanisms.
CO4: Construct and execute multithreaded Java programs by understanding thread creation, life cycle, and synchronization.
CO5: Design and implement GUI-based applications using Applets and AWT components.

Reference Books:

1. Java - The Complete Reference By **Herbert Schildt and Danny Coward**, 13th edition, McGraw-Hill in **2024**.

2. Java How to Program- Paul Deitel & Harvey Deitel, Pearson / Prentice Hall ,11th Edition ~2018

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech II Year I Semester

25BACDSLC03 ALGORITHM DESIGN AND ANALYSIS LABORATORY

L	T	P	C
0	0	2	1

Pre Requisites: Data Structures Laboratory

Course Description:

This course provides hands-on experience in designing, implementing, and analyzing fundamental algorithms and advanced data structures used in real-world computing applications. Students will develop programs for searching, sorting, graph processing, optimization, and combinatorial problem-solving using algorithmic paradigms such as Divide and Conquer, Greedy Method, Dynamic Programming and Backtracking. The course emphasizes practical implementation, performance analysis, and application of algorithms to industry-oriented scenarios including database management, network design, route optimization, resource allocation, scheduling, and decision-making problems.

Course Objectives:

1. Understand and implement fundamental data structures and algorithm design techniques for solving real-world computational problems.
2. Strengthen the ability to identify and apply the suitable algorithm strategy for the given real-world problem.
3. Analyze the time and space complexity of various algorithms and compare their performance for different problem scenarios.
4. Apply algorithmic paradigms such as Divide and Conquer, Greedy Method, Dynamic Programming, and Backtracking in practical applications.
5. Design efficient solutions for optimization and graph-related problems encountered in industry and research domains.

List of Experiments:

1. A banking system maintains customer account records that are frequently added and removed. Design an AVL tree to store account IDs from a file, support insertion/deletion of customer accounts, and generate a sorted report of active accounts into another file.
2. A database management system stores employee records on disk. Construct a B-Tree of order 5 to efficiently manage 100 employee IDs, supporting fast search, insertion of new employees, and deletion of resigned employees.
3. An e-commerce company receives thousands of product prices daily. Sort the prices using Quick Sort and Merge Sort. Analyze the performance for different dataset sizes to determine its suitability for real-time processing.
4. A cargo company must load goods into a truck with limited capacity. Select items based on value-to-weight ratio to maximize profit while utilizing available space effectively. Implement the Knapsack Problem using the Greedy method.
5. A navigation application needs to find the shortest route from a user's current location to all nearby destinations. Implement Dijkstra's algorithm to determine optimal travel paths.
6. Construct Minimum Spanning Tree:
7. A telecommunications company wants to connect multiple cities using fiber-optic cables with minimum installation cost. Use Prim's algorithm to design the network.

8. An electric utility company plans to connect substations with minimum wiring cost while ensuring all locations are connected. Apply Kruskal's algorithm to find the optimal network.
9. An airline company needs to determine the shortest travel distance between every pair of airports in its network for route planning and fare calculation. Find All Pairs Shortest Paths using Dynamic Programming method
10. A logistics company wants to identify the shortest routes from a central distribution center to all branch offices, considering possible negative route costs (discounts/tolls). Implement Single Source Shortest Paths using Dynamic Programming method.
11. A spacecraft has limited payload capacity. Choose scientific instruments to maximize research value without exceeding the weight limit. Each instrument can either be selected entirely or not selected at all. Write a program to solve 0/1 Knapsack problem Using Dynamic Programming.
12. A sales representative must visit multiple cities exactly once and return to the starting city while minimizing travel expenses and distance. Implement the Travelling Sales Person problem using Dynamic Programming.
13. A security agency wants to place surveillance cameras in a square facility such that no camera's monitoring path overlaps with another. Determine valid placements by implementing the N-Queens strategy.
14. A finance manager needs to identify combinations of investment options whose total investment exactly matches a target budget. Find all possible valid combinations. Design and implement using Backtracking to find all the possible subsets. For example, if S = {1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}.

Course Outcomes:

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

CO1: Implement and analyze tree-based data structures such as AVL Trees and B-Trees for efficient storage, retrieval, insertion, and deletion operations.

CO2: Develop and evaluate sorting and searching algorithms and compare their performance based on computational complexity and dataset characteristics.

CO3: Apply Greedy algorithms to solve optimization problems including Knapsack, Dijkstra's Shortest Path, Prim's MST, and Kruskal's MST.

CO4: Design and implement Dynamic Programming solutions for shortest path problems, 0/1 Knapsack, and Travelling Salesperson Problem.

CO5: Develop backtracking-based solutions for combinatorial problems such as N-Queens and Subset Sum.

Reference Books:

1. "Fundamentals of Data Structures in C++", Horowitz Ellis, Sahni Sartaj, Mehta, Dinesh, 2nd Edition, Universities Press, 2006.
2. "Computer Algorithms/C++", Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2nd Edition, University Press,2008.
3. "An introduction to Data Structures with Applications", Trembley & Sorenson, McGraw Hill Education, 2nd Edition,1997.

Mode of Evaluation: Continuous Internal Evaluation, Model Test 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

25BACDSTC05 DATA ENGINEERING

L	T	P	C
3	0	0	3

Pre-requisite Basic knowledge of programming and Database Management Systems (SQL).

Course Description:

This course introduces the fundamental concepts, principles, and practices of data engineering required designing, building and managing scalable data systems. Students will explore modern data architectures, data warehouses and data lakes, along with batch and streaming data processing techniques. The course also emphasizes data modeling, query optimization, data transformation, analytics enablement, and machine learning data pipelines.

Course Objectives:

This course enables students to

1. Describe the concepts, roles, and responsibilities associated with Data Engineering and its ecosystem.
2. Examine the stages of the Data Engineering Lifecycle and the technologies that support data movement and management.
3. Design robust data architectures and identify suitable data sources and integration mechanisms.
4. Implement appropriate storage, ingestion, and processing solutions for structured and unstructured data.
5. Evaluate and optimize data models, queries, transformations, and data-serving strategies for analytics and machine learning workloads.

UNIT I INTRODUCTION TO DATA ENGINEERING 9 hours

Introduction to Data Engineering: Definition, Data Engineering Life Cycle, Evolution of Data Engineer, Data Engineering Versus Data Science, Data Engineering Skills and Activities, Data Maturity, Data Maturity Model, Skills of a Data Engineer, Business Responsibilities, Technical Responsibilities, Data Engineers and Other Technical Roles.

UNIT II DATA ENGINEERING LIFECYCLE AND CORE COMPONENTS 9 hours

Data Engineering Life Cycle: Data Life Cycle Versus Data Engineering Life Cycle, Generation: Source System, Storage, Ingestion, Transformation, Serving Data. Major undercurrents across the Data Engineering Life Cycle: Security, Data Management, DataOps, Data Architecture, Orchestration, Software Engineering.

UNIT III DATA ARCHITECTURES 9 hours

Designing Good Data Architecture: Enterprise Architecture, Data Architecture, Principles of Good Data Architecture, Major Architecture Concepts. Data Generation in Source Systems: Sources of Data, Files and Unstructured Data, APIs, Application Databases (OLTP), OLAP, Change Data Capture, Logs, Database Logs, CRUD, Source System Practical Details.

UNIT IV DATA INGESTION 9 hours

Storage: Raw Ingredients of Data Storage, Data Storage Systems, Data Engineering Storage Abstractions, Data warehouse, Data Lake, Data Lake house. Ingestion: Data Ingestion, Key

Engineering considerations for the Ingestion Phase, Batch Ingestion Considerations, Message and Stream Ingestion Considerations, Ways to Ingest Data.

UNIT V DATA TRANSFORMATION AND REVERSE ETL 9 hours

Queries, Modeling and Transformation: Queries, Life of a Query, Query Optimizer, Queries on Streaming Data, Data Modelling, Modeling Streaming Data, Transformations, Streaming Transformations and Processing. Serving Data for Analytics, Machine Learning and Reverse ETL: General Considerations for serving Data, Business Analytics, Operational Analytics, Embedded Analytics, ways to serve data for analytics and ML, Reverse ETL.

Course Outcomes:

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

CO1: Understand data engineering fundamentals, differentiate from data science, and identify key data engineer roles and responsibilities.

CO2: Grasp the data engineering lifecycle, including security, management, and operational aspects.

CO3: Design effective data architectures based on data sources, models, and architectural principles.

CO4: Select optimal storage and ingestion methods for efficient data pipelines.

CO5: Master querying, modeling, and transformation techniques to deliver valuable data products.

Text Book(s)

1. Joe Reis, Matt Housley, "Fundamentals of Data Engineering", O Reilly Media Inc, 1st Edition, 2022.
2. Paul Crickard, Data Engineering with Python: Work with Massive Datasets to Design Data Models and Automate Data Pipelines Using Python, Packt Publishing, 1st Edition, 2021.

Reference Books

1. James Densmore, Data Pipelines Pocket Reference: Moving and Processing Data for Analytics, O'Reilly Media, Inc., 1st Edition, 2021.
2. Andreas Kretz, "The Data Engineering Cookbook", The Data Engineering Academy, 1st Edition, 2019.
3. Ralph Kimball, Margy Ross, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Wiley, 3rd Edition, 2013.

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

B. Tech II Year II Semester

25BACDSTC06 APPLIED MACHINE LEARNING

L	T	P	C
3	0	0	3

Pre-requisite: Python for Data Science, Linear Algebra

Course Description:

This course introduces the fundamental concepts of Machine Learning, focusing on data preprocessing, supervised learning, unsupervised learning, neural networks, and reinforcement learning techniques. It covers machine learning algorithms, model evaluation, clustering, dimensionality reduction, neural network architectures, and intelligent decision-making systems. The course enables students to design and apply machine learning models for solving real-world data-driven problems.

Course Objectives:

This course enables students to:

1. Understand the foundations, workflow, and applications of Machine Learning.
2. Apply supervised learning algorithms for regression and classification problems.
3. Analyze and evaluate machine learning models using suitable performance metrics.
4. Understand unsupervised learning and neural network architectures.
5. Develop reinforcement learning models for intelligent decision-making applications.

UNIT I OVERVIEW OF MACHINE LEARNING ALGORITHMS AND DATA PREPROCESSING **9 hours**

Introduction to Machine Learning – History and Applications of Machine Learning – Types of Machine Learning: Supervised, Unsupervised, and Reinforcement Learning – Data Preprocessing: Types of Data, Handling Missing Data, Data Scaling: Standardization and Normalization, Encoding Categorical Variables, Train-Test Split.

UNIT II SUPERVISED LEARNING MODELS AND EVALUATION METRICS **9 hours**

Linear Regression – Loss Functions – Gradient Descent – Multiple Linear Regression – Classification: K-Nearest Neighbor (KNN), Decision Tree, Random Forest – Model Evaluation Metrics: Accuracy, Precision, Recall, F1 Score, Confusion Matrix – Overfitting and Underfitting – Cross Validation – Hyperparameter Tuning: Grid Search and Random Search.

UNIT III UNSUPERVISED LEARNING **9 hours**

Introduction to Unsupervised Learning – K-Means Clustering – Hierarchical Clustering – Cluster Evaluation – Dimensionality Reduction – Principal Component Analysis (PCA) – Applications of Unsupervised Learning.

UNIT IV PERCEPTRON AND FEED FORWARD NEURAL NETWORKS **9 hours**

Perceptron: logistic regression, sigmoid activation function, binary classification, error backpropagation algorithm, feed-forward neural network, universal approximation theorem.

UNIT V REINFORCEMENT LEARNING 9 hours

Introduction to Reinforcement Learning – Agent, Environment, State, Action, Reward – Markov Decision Process (MDP) – Q-Learning – Bellman Equation – Exploration vs Exploitation – Policy and Value Functions – Applications of Reinforcement Learning in Robotics, Gaming, Recommendation Systems, and Autonomous Systems.

Course Outcomes:

After completing this course, students will be able to:

CO1: Explain machine learning concepts, workflow, and data preprocessing techniques.

CO2: Apply supervised learning algorithms for regression and classification problems.

CO3: Implement unsupervised learning techniques for clustering and dimensionality reduction.

CO4: Analyze perceptron and feed forward neural network architectures for predictive modeling.

CO5: Develop reinforcement learning models for intelligent decision-making applications.

Text Book(s)

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
2. Pattern Recognition and Machine Learning, New York, NY, USA: Springer, 2006.
3. Introduction to Machine Learning with Python, Sebastopol, CA, USA: O'Reilly Media, 2016.
4. Reinforcement Learning: An Introduction, MIT Press, 2018.

Reference Books

1. Deep Learning, Cambridge, MA, USA: MIT Press, 2016.
2. Machine Learning, New York, NY, USA: McGraw-Hill Education, 1997.
3. Machine Learning: A Probabilistic Perspective, Cambridge, MA, USA: MIT Press, 2012.

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

B. Tech II Year II Semester

25BACDSTC07 OPERATING SYSTEMS FUNDAMENTALS

L	T	P	C
3	0	0	3

Pre-requisite: Basic knowledge of Computer Organization, Programming, and Data Structures.

Course Description:

This course introduces the fundamental concepts of Operating Systems, including process management, memory management, file systems, synchronization, deadlocks, storage management, and distributed systems. The course enables students to understand the design and functioning of modern operating systems and their role in efficient resource management and system performance.

Course Objectives:

This course enables students to:

1. Understand the structure, functions, and services of operating systems.
2. Analyze process management, scheduling, synchronization, and deadlock handling techniques.
3. Understand memory management strategies and virtual memory concepts.
4. Apply file system and storage management concepts in operating system design.
5. Understand distributed operating systems and system protection mechanisms.

UNIT I INTRODUCTION TO OPERATING SYSTEMS AND PROCESS MANAGEMENT **9 hours**

Operating Systems –Definition- Types- Functions -Abstract view of OS- System Structures – System Calls- Virtual Machines –Process Concepts –Threads –Multithreading.

UNIT II PROCESS SCHEDULING, SYNCHRONIZATION, AND DEADLOCKS **9 hours**

Process Scheduling- Process Co-ordination –Synchronization –Semaphores –Monitors Hardware Synchronization –Deadlocks –Methods for Handling Deadlocks

UNIT III MEMORY MANAGEMENT AND VIRTUAL MEMORY **9 hours**

Memory Management Strategies –Contiguous and Non-Contiguous allocation –Virtual memory Management –Demand Paging- Page Placement and Replacement Policies

UNIT IV FILE SYSTEMS, STORAGE MANAGEMENT, AND SECURIT **9 hours**

File System –Basic concepts - File System design and Implementation –Case Study: Linux File Systems - Mass Storage Structure –Disk Scheduling –Disk Management –I/O Systems- System Protection and Security.

UNIT V DISTRIBUTED OPERATING SYSTEMS **9 hours**

Distributed Systems –Distributed operating systems –Distributed file systems –Distributed Synchronization.

Course Outcomes:

After completing this course, students will be able to:

CO1: Explain the structure, functions, and services provided by operating systems.

CO2: Analyze process scheduling, synchronization, and deadlock handling techniques.

CO3: Apply memory management and virtual memory concepts for efficient resource utilization.

CO4: Understand file system organization, storage management, and system security mechanisms.

CO5: Explain distributed operating systems and distributed synchronization techniques.

Text Book(s)

1. Silberschatz, Galvin, Gagne, "Operating System Concepts", John Wiley and Sons, 9th edition, 2013

Reference Books

1. William Stallings, “Operating Systems –Internals and Design Principles”, 8/E, Pearson Publications, 2014
2. Andrew S. Tanenbaum, “Modern Operating Systems”, 4/E, Pearson Publications, 2014

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

B. Tech II Year II Semester

25BACDSLC04 DATA ENGINEERING LABORATORY

L	T	P	C
0	0	2	1

Pre Requisites: **25BACDSTC02, 25BACDSSC01**

Course Description:

This laboratory course provides hands-on experience in designing, implementing, and monitoring modern data engineering pipelines. Students gain practical exposure to data ingestion, storage, workflow orchestration, data transformation, and visualization using industry-standard tools such as Apache NiFi, Apache Airflow, PostgreSQL, Elasticsearch, Kibana, Apache Spark, and PySpark. The course emphasizes building scalable ETL pipelines, integrating heterogeneous data sources, and monitoring data workflows for reliable and efficient data processing.

Course Objectives:

This course enables students to

1. Understand the installation, configuration, and usage of modern data engineering tools and platforms.
2. Develop skills in relational and NoSQL database operations using Python.
3. Design and implement data ingestion and ETL pipelines using Apache NiFi and Apache Airflow.
4. Perform large-scale data processing and transformation using Apache Spark and PySpark.
5. Build end-to-end data engineering workflows, monitor pipeline execution, and visualize processed data using Kibana.

List of Experiments:

1. Installing and configuring Apache NiFi, Apache Airflow.
2. Installing and configuring Elasticsearch, Kibana, PostgreSQL, pgAdmin.
3. Database Foundations (PostgreSQL & pgAdmin)
   a) Creating databases and tables
   b) Reading and writing data to databases in Python
   i) Connecting Python with PostgreSQL
   ii Performing basic CRUD operations
   c) Inserting and extracting relational data in Python
   d) Reading from and writing to a NoSQL database (e.g., MongoDB) using Python.
4. Data Ingestion & Flow (Apache NiFi)
   a) NiFi processors for handling files, File ingestion and routing
   b) Database processors in NiFi, Connecting NiFi to PostgreSQL.
   c) Building database pipelines in NiFi.
5. Version Control in NiFi
   a) Installing and configuring the NiFi Registry
   b) Using the Registry in NiFi
   c) Versioning your data pipelines
   d) Using git-persistence with the NiFi Registry.
6. Workflow Orchestration (Apache Airflow)
   a) Processing files in Airflow, File-based DAGs

b) Databases in Airflow, Database tasks in DAGs
c) Building database pipelines in Airflow
d) Cleaning data using Airflow.

7. Core Data Handling & Cleaning (Python)
   a) Reading and writing files (CSV, JSON) in Python
   b) Performing exploratory data analysis in Python
   c) Handling common data issues using pandas.
   d) Introduction to PySpark and SparkSession.
   e) Creating and manipulating PySpark DataFrames.
   f) Reading CSV and JSON files using PySpark.
8. Building Reliable Data Pipelines
   a) Building the data pipeline by integrating NiFi, Airflow, and PostgreSQL.
   b) Staging and validating data.
   c) Installation and configuration of Apache Spark and PySpark.
   d) Performing data transformation and aggregation using PySpark.
   e) Reading from and writing to PostgreSQL using PySpark.
   f) Designing and implementing an end-to-end ETL pipeline using PySpark.
9. Search, Analytics & Visualization
   a) Building a Kibana Dashboard for Visual analytics.
10. Monitoring the Data Pipeline.
    a) Monitoring NiFi in the GUI
    b) Monitoring NiFi using processors
    c) Monitoring NiFi with Python and the REST API.
    d) Monitoring Apache Airflow DAG execution and logs.

Course Outcomes:

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

CO1: Install, configure, and utilize data engineering tools including Apache NiFi, Airflow, PostgreSQL, Elasticsearch, Kibana, Apache Spark, and PySpark.

CO2: Design and implement database operations, data ingestion workflows, and ETL pipelines using Python, PostgreSQL, and Apache NiFi.

CO3: Develop workflow orchestration solutions using Apache Airflow for automating data processing tasks.

CO4: Apply PySpark for distributed data processing, transformation, and implementation of scalable end-to-end data pipelines.

CO5: Monitor data pipelines, evaluate workflow performance, and create interactive dashboards for data visualization using Kibana.

Reference Books:

1. Paul Crickard, Data Engineering with Python: Work with Massive Datasets to Design Data Models and Automate Data Pipelines Using Python, Packt Publishing, 1st Edition, 2021.

2. Seema Acharya, Subhasini Chellappan, *Big Data and Analytics*, Wiley India, 2nd Edition.2019.

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech II Year II Semester

25BACDSLC05 APPLIED MACHINE LEARNING LABORATORY

L	T	P	C
0	0	2	1

Pre Requisites: Python Programming, Fundamental Knowledge of Statistics

Course Description:

This laboratory course provides hands-on experience in implementing Machine Learning algorithms using Python-based tools and libraries. Students learn data preprocessing, supervised and unsupervised learning, model evaluation, neural networks, and reinforcement learning techniques using real-world datasets for predictive analysis and intelligent decision making.

Course Objectives:

The objectives of this laboratory course are to enable students to:

1. Provide hands-on experience in data preprocessing and feature engineering techniques.
2. Develop practical skills in implementing supervised and unsupervised learning algorithms.
3. Enable students to evaluate machine learning models using appropriate performance metrics.
4. Train students to develop neural network and reinforcement learning models for intelligent systems.
5. Enhance analytical and problem-solving skills for real-world machine learning applications.

List of Experiments:

1. Perform data preprocessing techniques such as handling missing values, feature scaling, normalization, encoding categorical variables, and train-test split for a given dataset.
2. Detect and handle outliers using statistical and visualization methods for data cleaning.
3. Build Linear Regression and Multiple Linear Regression models to analyze relationships between dependent and independent variables.
4. Implement K-Nearest Neighbor (KNN), Decision Tree, and Random Forest algorithms for classification and compare their performance using evaluation metrics.
5. Apply model evaluation techniques including Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.
6. Perform Cross Validation and hyperparameter tuning using Grid Search and Random Search methods.
7. Implement K-Means clustering algorithm to group similar data samples in a dataset.
8. Perform dimensionality reduction using Principal Component Analysis (PCA) and visualize transformed features.
9. Build and train Perceptron and Feed Forward Neural Network models for binary and multi-class classification tasks.
10. Implement Q-Learning for simple reinforcement learning environments and analyze agent behavior.
11. Develop reinforcement learning models for applications such as gaming, path finding, and recommendation systems.
12. Visualize and analyze machine learning and neural network models using suitable visualization and performance analysis tools.

Tools / Software: Python, Jupyter Notebook, Google Colab, Anaconda, Scikit-learn, TensorFlow, Keras, Netron, NumPy, Pandas, Matplotlib, OpenCV

Course Outcomes:

After completing the laboratory course, students will be able to:

CO1: Perform preprocessing and feature engineering techniques on machine learning datasets.
CO2: Implement supervised learning algorithms for regression and classification problems.
CO3: Apply clustering and dimensionality reduction techniques for unsupervised learning tasks.
CO4: Develop and evaluate perceptron and feed forward neural network models using appropriate performance metrics.
CO5: Build reinforcement learning models for intelligent decision-making applications.

Text Books:

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed., O'Reilly Media, 2022.
2. Introduction to Machine Learning with Python, O'Reilly Media, 2016.

Reference Books:

1. Joel Grus, *Data Science from Scratch*, O'Reilly, 2019
2. Python Machine Learning, Packt Publishing, 2019.

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech II Year II Semester

25BACDSLC06 OPERATING SYSTEMS FUNDAMENTALS LABORATORY

L	T	P	C
0	0	2	1

Pre Requisites: Basics of Programming in C/Python and Computer Organization.

Course Description:

This laboratory course provides practical exposure to operating system concepts including process management, CPU scheduling, synchronization, deadlocks, memory management, file systems, and disk scheduling. Students implement various operating system algorithms and system-level concepts through programming exercises and simulations.

Course Objectives:

This laboratory course enables students to:

1. Understand process management and CPU scheduling algorithms through practical implementation.
2. Develop synchronization and deadlock handling mechanisms using operating system concepts.
3. Implement memory management and page replacement algorithms.
4. Apply file allocation and disk scheduling techniques in operating systems.
5. Develop practical skills in inter-process communication and multithreading concepts.

List of Experiments:

1. Implement Linux/UNIX commands for file and directory management.
2. Write a program using system calls such as fork(), exec(), wait(), and exit() for process creation and management.
3. Implement multithreading using POSIX threads (Pthreads).
4. Implement CPU scheduling algorithms:
    a. First Come First Serve (FCFS)
    b. Shortest Job First (SJF)
    c. Round Robin (RR)
5. Implement Priority Scheduling algorithm for process scheduling.
6. Implement Producer–Consumer problem using semaphores.
7. Implement Reader–Writer problem using synchronization techniques.
8. Implement Banker's Algorithm for deadlock avoidance.
9. Implement deadlock detection algorithm.
10. Implement contiguous memory allocation techniques:
    a. First Fit
    b. Best Fit
    c. Worst Fit
11. Implement page replacement algorithms:
    a. FIFO
    b. LRU
    c. Optimal Page Replacement
12. Implement demand paging technique.
13. Implement file allocation methods:
    a. Sequential Allocation
    b. Indexed Allocation
    c. Linked Allocation

14. Implement disk scheduling algorithms:
    a. FCFS
    b. SSTF
    c. SCAN
    d. C-SCAN
15. Implement inter-process communication using pipes and shared memory.

Course Outcomes:

After completing this laboratory course, students will be able to:

CO1: Implement process creation and CPU scheduling algorithms using system calls.
CO2: Apply synchronization techniques and deadlock handling algorithms for process coordination.
CO3: Implement memory management and page replacement algorithms for efficient resource utilization.
CO4: Develop programs for file allocation methods and disk scheduling techniques.
CO5: Implement inter-process communication and multithreading concepts in operating systems.

Text Books:

1. Operating System Concepts, 9th ed., John Wiley & Sons, 2013.

2. Modern Operating Systems, 4th ed., Pearson Publications, 2014.

Reference Books:

1. Operating Systems: Internals and Design Principles, 8th ed., Pearson Publications, 2014.

2. Unix Concepts and Applications, 4th ed., McGraw-Hill Education.

3. Advanced Programming in the UNIX Environment, 3rd ed., Pearson Education.

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

Skill Enhancement Courses

Skill Enhancement Course - I

25BACDSSC01 PYTHON FOR DATA SCIENCE

L	T	P	C
1	0	2	2

Pre-requisite: Basic understanding of mathematics, statistics, and programming concepts.

Course Description:

This course introduces Python programming concepts and their applications in Data Science. Students learn Python fundamentals, control structures, functions, data structures, numerical computing using NumPy, data preprocessing and analysis using Pandas, and data visualization techniques using Matplotlib and Seaborn for solving real-world data-driven problems.

Course Objectives:

This course enables students to:

1. Introduce Python programming fundamentals and control structures for Data Science applications.
2. Understand functions, strings, and modular programming concepts using Python.
3. Apply Python data structures for organizing, storing, and processing data.
4. Perform numerical computations and statistical analysis using NumPy.
5. Analyze, preprocess, and interpret datasets using Pandas tools.

UNIT I PYTHON PROGRAMMING FUNDAMENTALS 6 hours

Introduction to Python Environment, Variables and Data Types, Operators, Input and Output Operations, Conditional Statements, Looping Statements.

1. A supermarket wants to calculate the total bill of customers with discounts based on purchase amount. Develop a program using conditional statements and operators to calculate the final bill amount after applying discounts based on purchase amount.
   Sample Input:
   Purchase Amount = 5000
   Expected Output:
   Discount = 10%
   Final Bill = 4500
   Test Cases:

Test Case	Input Amount	Expected Output
TC1	1000	No Discount
TC2	3000	5% Discount
TC3	5000	10% Discount

2. A college wants to identify students who passed or failed based on marks entered by the user. Develop a program using looping and conditional statements to calculate total marks, average marks, grades, and pass/fail status for multiple students.
   Sample Input:
   Marks = 78
   Expected Output:
   Result = Pass
   Grade = First Class

Test Cases:

Test Case	Input	Expected Output
TC1	25	Fail
TC2	40	Pass
TC3	78	First Class

3. A traffic department wants to calculate penalties for vehicle speed violations. Develop a program using loops and conditional statements to categorize vehicles based on speed limits and generate fine details for traffic violations.
   Sample Input:
   Vehicle Speed = 92 km/h
   Expected Output:
   Fine Category = Heavy Fine
   Test Cases:

Test Case	Input	Expected Output
TC1	45 km/h	No Fine
TC2	70 km/h	Moderate Fine
TC3	92 km/h	Heavy Fine

UNIT II FUNCTIONS AND STRINGS IN PYTHON 6 hours

Functions, Function Definition and Calling, Function Arguments, Return Values, Lambda Expressions, String Operations, String Indexing and Slicing, Built-in String Methods.

4. A company wants to calculate employee salaries using reusable functions. Develop a program using functions and lambda expressions to calculate gross salary, deductions, bonuses, and net salary for employees.
   Sample Input:
   Basic Salary = 30000
   Expected Output:
   Gross Salary Calculated
   Net Salary Calculated
   Test Cases:

Test Case	Input	Expected Output
TC1	20000	Salary Calculated
TC2	30000	Allowances Added
TC3	50000	Tax Deducted

5. A mobile service provider wants to analyze customer names and phone details. Develop a program using string operations to validate mobile numbers, extract customer information, search substrings, and format customer details.

Sample Input:
Customer Name = Ravi Kumar
Expected Output:
First Name = Ravi
Last Name = Kumar

Test Cases:

Test Case	Input	Expected Output
TC1	Ravi Kumar	Name Split
TC2	9876543210	Valid Number
TC3	DataScience	Substring Extracted

6. A power distribution company wants to calculate electricity bills based on customer electricity usage. Develop a Python program using functions to calculate the total electricity bill by applying tariff rates according to the number of units consumed.
   Sample Input:
   Units Consumed = 250 units
   Expected Output:
   Bill Amount Calculated
   Total Electricity Bill = ₹1250
   Test Cases:

Test Case	Input (Units Consumed)	Expected Output
TC1	80 units	Low Usage Bill Calculated
TC2	180 units	Medium Usage Bill Calculated
TC3	350 units	High Usage Bill Calculated

UNIT III PYTHON DATA STRUCTURES 6hours

Lists, Tuples, Sets, Dictionaries, List Operations, Dictionary Operations, Searching and Updating Elements, Iterating Data Structures, Nested Collections.

7. Develop a library management application using lists, tuples, sets, and dictionaries to store, search, update, and display book details.
8. Write a Python program to categorize shopping products based on price ranges and stock availability using Python data structures.
9. Develop a student record management program us

UNIT IV NUMERICAL COMPUTING USING NUMPY 6 hours

Introduction to NumPy, NumPy Arrays, Array Creation, Indexing and Slicing, Array Reshaping, Mathematical Operations on Arrays, Statistical Functions, Random Number Generation.

10. Develop a Python program to analyze temperature data using NumPy arrays, indexing, slicing, reshaping, and statistical operations.
11. Write a Python program to perform arithmetic and statistical operations on bank transaction datasets using NumPy.
12. Develop a Python program to generate random sample data and compute statistical measures such as mean, median, variance, and standard deviation.

UNIT V DATA ANALYSIS USING PANDAS 6 hours

Introduction to Pandas, Series and DataFrames, Data Selection and Filtering, Data Cleaning, Handling Missing Values, Removing Duplicate Records, Exploratory Data Analysis, Statistical Summaries and Correlation Analysis.Import a dataset into SPSS and generate descriptive statistics.

13. Develop a Python program to create and analyze student performance datasets using Pandas Series and DataFrames.
14. Write a Python program to preprocess an e-commerce dataset by handling missing values, duplicate records, and invalid data.
15. Develop a Python program to perform Exploratory Data Analysis (EDA) and correlation analysis on real-world datasets using Pandas.

Course Outcomes:

Upon successful completion of the course, students will be able to

CO1: Explain Python programming fundamentals, operators, and control structures for solving computational problems.

CO2: Develop Python programs using functions, lambda expressions, and string processing techniques.

CO3: Apply Python data structures such as lists, tuples, sets, and dictionaries for data handling applications.

CO4: Perform numerical and statistical computations using NumPy arrays and built-in functions.

CO5: Analyze and preprocess real-world datasets using Pandas Series, DataFrames, and exploratory data analysis techniques.

Text Book(s)

1. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
2. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd ed. Needham, MA, USA: Green Tea Press, 2015.
3 Eric Matthes, Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 3rd ed. San Francisco, CA, USA: No Starch Press, 2023.

Reference Books

1. John V. Guttag, Introduction to Computation and Programming Using Python: With Application to Computational Modeling and Understanding Data, 3rd ed. Cambridge, MA, USA: MIT Press, 2021.
2. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
3. Y. Daniel Liang, Introduction to Programming Using Python, 1st ed. Boston, MA, USA: Pearson Education, 2013.

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

Skill Enhancement Course - II

25BACDSSC02 FULL STACK DEVELOPMENT

L	T	P	C
1	0	2	2

Pre-requisite: Programming Concepts

Course Description:

This course provides practical training in Full Stack Web Development using modern frontend, backend, and database technologies. Students learn to design responsive web pages, develop dynamic applications, create REST APIs, manage databases, and deploy full-stack applications for real-world use cases.

Course Objectives:

This course enables students to:

1. Develop responsive and interactive web pages using frontend technologies.
2. Understand backend development and server-side programming concepts.
3. Design and manage databases for web applications.
4. Develop RESTful APIs and integrate frontend with backend systems.
5. Build and deploy complete full-stack web applications.

UNIT I FRONTEND DEVELOPMENT FUNDAMENTALS 6 hours

HTML Basics, HTML Forms, CSS Styling, Responsive Design, JavaScript Basics, DOM Manipulation, Events, Form Validation.

A college wants to build a basic student information portal using frontend technologies.

1. Create a responsive student registration form using HTML and CSS.
2. Develop a webpage with navigation menus and responsive layouts.
3. Implement form validation using JavaScript.
4. Create dynamic webpage content using DOM manipulation.
5. Design a simple portfolio webpage using HTML, CSS, and JavaScript.

UNIT II ADVANCED FRONTEND DEVELOPMENT 6 hours

Introduction to React – Components – Props and State – Event Handling – React Hooks – Routing – Fetching API Data – Responsive UI Design.

A company wants to create interactive frontend applications for users.

1. Create reusable React components for a student dashboard.
2. Develop a counter and calculator application using React state management.
3. Implement routing between multiple pages in React.
4. Fetch and display data from a public API using React.
5. Design a responsive user interface using React components.

UNIT III BACKEND DEVELOPMENT WITH NODE.JS AND EXPRESS.JS 6 hours

Introduction to Node.js – Node Package Manager (NPM) – Express.js Framework – REST API Development – Middleware – Routing – CRUD Operations – API Testing using Postman.

A college administration wants to manage student records through backend services.

1. Create a Node.js server application.
2. Develop REST APIs for student data management.
3. Implement CRUD operations using Express.js.
4. Test APIs using Postman.
5. Develop middleware for request handling and authentication.

UNIT IV DATABASE MANAGEMENT AND INTEGRATION 6 hours

Introduction to Databases – MongoDB Basics – Collections and Documents – CRUD Operations in MongoDB – Database Connectivity – Integration of Frontend and Backend.

A university wants to store and retrieve student records using databases.

1. Create and manage collections in MongoDB.
2. Perform insert, update, delete, and retrieval operations.
3. Connect MongoDB with Node.js applications.
4. Integrate frontend forms with backend database operations.
5. Develop a complete student management module with database support.

UNIT V Full Stack Application Development and Deployment 6 hours

Authentication and Authorization – Session Management – JWT Basics – Error Handling – Full Stack Project Structure – Deployment Basics – Version Control using Git and GitHub.

A startup company wants to deploy a complete full-stack application.

1. Implement user login and registration functionality.
2. Develop authentication using JWT tokens.
3. Handle errors and exceptions in full-stack applications.
4. Upload project code using Git and GitHub.
5. Develop and deploy a mini full-stack web application.

Suggested Tools: Visual Studio Code ,Node.js ,Express.js ,MongoDB ,React ,Postman ,GitHub

Course Outcomes:

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

CO1: Develop responsive web pages using frontend technologies.
CO2: Build interactive user interfaces using React.
CO3: Develop backend applications and REST APIs using Node.js and Express.js.
CO4: Design and integrate MongoDB databases with web applications.
CO5: Build and deploy complete full-stack web applications.

Text Book(s)

1. Web Development with Node and Express, O'Reilly Media.
2. MongoDB: The Definitive Guide, O'Reilly Media.
3. Learning React, O'Reilly Media.

Reference Books

1. Full Stack React Projects, Packt Publishing.
2. JavaScript: The Good Parts, O'Reilly Media.
3. Pro MERN Stack, Apress Publications.

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