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

Estd 1998

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
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 & ENGINEERING

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(Deemed to be University)

MADANAPALLE

B. Tech Four Year Curriculum Structure

Branch: COMPUTER SCIENCE & ENGINEERING

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	AEC	25BAENGTC01	Professional Communication	2	0	0	2	2
2	HSMC	25BAHUMTC01	Human Values and Professional Ethics	2	0	0	2	2
3	BSC	25BAMATTC02	Calculus for Computer Science	3	0	0	3	3
4	BSC	25BAPHYTC03	Modern Physics for Quantum Technologies	3	0	0	3	3
5	ESC	25BACSETC01	C Programming for Problem Solving	3	0	0	3	3
6	PCC	25BACSETC02	Fundamentals of Web Technology	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	25BACSELC01	C Programming for Problem Solving Laboratory	0	0	2	2	1
10	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	BSC	25BAMATTC05	Linear Algebra	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	25BACSETC03	Data Structures and Algorithms	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	25BACOMLC01	Engineering Skills Laboratory	0	0	2	2	1
9	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	25BAMATTC12	Discrete Mathematical Structures	3	0	0	3	3
2	ESC	25BACOMTC01	Computer Organization and Architecture	3	0	0	3	3
3	PCC	25BACSETC04	Database Management Systems	3	0	0	3	3
4	PCC	25BACSETC05	Object Oriented Programming	3	0	0	3	3
5	PCC	25BACSETC06	Analysis of Algorithms	3	0	0	3	3
6	PCC	25BACSELC05	Database Management Systems Laboratory	0	0	2	2	1
7	PCC	25BACSELC06	Object Oriented Programming Laboratory	0	0	2	2	1
8	PCC	25BACSELC07	Analysis of Algorithms 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**	**10**	**26**	**20**

II Year II 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	PCC	25BACSETC07	Machine Learning and Applications	3	0	0	3	3
3	PCC	25BACSETC08	Data Communication and Computer Networks	3	0	0	3	3
4	PCC	25BACSETC09	Operating Systems	3	0	0	3	3
5	OE		**Open Elective – I** *(Refer ANNEXURE - II)*	3	0	0	3	3
6	PCC	25BACSELC08	Machine Learning and Applications Laboratory	0	0	2	2	1
7	PCC	25BACSELC09	Data Communication and Computer Networks Laboratory	0	0	2	2	1
8	PCC	25BACSELC10	Operating Systems 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 Startups	3	0	0	3	3
2	PCC	25BACSETC10	Automata Theory and Compiler Design	3	0	0	3	3
3	PCC	25BACSETC11	Artificial Intelligence	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	25BACSELC11	Compiler Design Laboratory	0	0	2	2	1
7	PCC	25BACSELC12	Artificial Intelligence Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – III** *(Refer ANNEXURE - IV)*	1	0	2	3	2
9	INT	25BACSEIC01	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		Managerial Economics and Accounting	3	0	0	3	3
2	PCC	25BACSETC12	Introduction to Quantum Computing	2	0	0	2	2
3	PCC	25BACSETC13	Cloud Computing	3	0	0	3	3
4	PCC	25BACSETC14	Cryptography and Information Security	2	0	0	2	2
5	DE		**Discipline Elective – II** *(Refer ANNEXURE - III)*	3	0	0	3	3
6	OE		**Open Elective – III (MOOC's)** *(Refer ANNEXURE - II)*	3	0	0	3	3
7	PCC	25BACSELC13	Cloud Computing Laboratory	0	0	2	2	1
8	PCC	25BACSELC14	Cryptography and Information Security 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	25BACSETC15	Internet of Things	3	0	0	3	3
2	PCC	25BACSETC16	Data Science and Business Intelligence	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	25BACSELC15	Data Science and Business Intelligence Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – IV** *(Refer ANNEXURE – IV)*	1	0	2	3	2
9	INT	25BACSEIC02	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	25BACSEPC01	Capstone Project	0	0	28	28	14
2	INT	25BACSEIC03	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.	25BACSEDM01	Big Data Computing
2.	25BACSEDM02	Privacy and Security in Online Social Media
3.	25BACSEDM03	Data Mining
4.	25BACSEDM04	Real-Time Systems
5.	25BACSEDM05	Object Oriented System Development Using UML, Java And Patterns
Any advanced Course offered by SWAYAM NPTEL can be appended in future.		

Discipline Elective – II		
S. No.	**Course Code**	**Course Title**
1.	25BACSEDC01	Linux System Administration and Automation
2.	25BACSEDC02	Cloud Security
3.	25BACSEDC03	Artificial Intelligence
4.	25BACSEDC04	Embedded Systems Design
5.	25BACSEDC05	Deep Learning
Any advanced courses can be appended in future.		

Discipline Elective – III		
S. No.	**Course Code**	**Course Title**
1.	25BACSEDC06	Software Project Management
2.	25BACSEDC07	Software Defined Networking
3.	25BACSEDC08	Natural Language Processing
4.	25BACSEDC09	Distributed Operating System
5.	25BACSEDC10	Generative AI
Any advanced courses can be appended in future.		

Discipline Elective – IV		
S. No.	**Course Code**	**Course Title**
1.	25BACSEDC11	Data Centre and Network Engineering
2.	25BACSEDC12	Blockchain Technology
3.	25BACSEDC13	Computer Vision
4.	25BACSEDC14	Data Centre Architecture
5.	25BACSEDC15	Agentic AI
Any advanced courses can be appended in future.		

Discipline Elective –V		
S. No.	**Course Code**	**Course Title**
1	25BACSEDC16	Agile methodologies
2	25BACSEDC17	Data Centre Infrastructure Management
3	25BACSEDC18	Augmented Reality and Virtual Reality
4	25BACSEDC19	Cyber Physical Systems
5	25BACSEDC20	Responsible AI
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.	25BACSESC01	Data Science Using Python
Any Courses can be appended in future.		

Skill Enhancement Course – II		
S. No.	**Course Code**	**Course Title**
1.	25BACSESC02	Full Stack Development
Any Courses can be appended in future.		

Skill Enhancement course – III		
Sl. No.	**Course Code**	**Course Title**
1.	25BACSESC03	Prompt Engineering
Any Courses can be appended in future.		

Skill Enhancement Course – IV		
Sl. No.	**Course Code**	**Course Title**
1.	25BACSESC04	NoSQL
Any Courses can be appended in future.		

I Year I Semester

B. Tech I Year I 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 I 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 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

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 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

25BACSETC02 FUNDAMENTALS OF WEB TECHNOLOGY

L	T	P	C
3	0	0	3

Pre-requisite None

Course Description:

Fundamentals of Web Technology provides a foundational understanding of web technologies and the principles of user interface design. It covers the creation of structured, visually appealing, and interactive web pages using HTML, CSS, and JavaScript, along with modern HTML5 and CSS3 features for responsive and accessible design.

Course Objectives:

This course enables students to

1. Understand client/server architecture, web server/application server roles, and HTML document structure.
2. Construct web pages using tags, links, tables, forms, and HTML5 elements like media and graphics.
3. Apply various CSS properties to style borders, backgrounds, text, fonts, and establish responsive layouts.
4. Validate Java Script code event handling, and manipulating DOM elements
5. Examine UI scenarios and recommend suitable semantic tags, responsive layouts, media queries, and CSS animations for improved usability and accessibility.

UNIT I INTRODUCTION TO WEB & HTML BASICS 9 hours

Introduction to Web – Client/Server Architecture – Web Server – Application Server – HTML Basics – HTML Tags & Attributes – Adding Web Links & Images – Creating Tables – Creating Forms – HTML5 Elements (Section, Article, Header, Footer) – Working with Audio & Video

UNIT II CASCADING STYLE SHEETS (CSS) 9 hours

Introduction to CSS – Features & Advantages – Types of CSS (Inline, Internal, External) – Syntax, Selectors & Properties – Implementing Borders – Backgrounds (Colors, Images, Gradients) – CSS3 Features – Text Effects – Fonts & @font-face – CSS Box Model – Page Layouts with CSS (Float, Position, Flexbox) – Simple Navigation Bar Styling

UNIT III JAVASCRIPT FOR WEB PAGES 9 hours

Introduction to JavaScript – Features & Syntax – Variables & Data Types – Operators – Functions & Scope – DOM (Document Object Model) – Access & Manipulate Elements – Events & Event Handling – Form Validations – Alert, Confirm & Prompt – Loops & Conditional Statements – Arrays & Objects – Simple Animations

UNIT IV ADVANCED HTML5 & CSS3 FOR UI DESIGN 9 hours

HTML5 Forms (New Input Types, Validation) – Semantic Tags for Accessibility – Responsive Web Design Basics – Media Queries – Mobile-Friendly Layouts – CSS3 Transitions & Transformations – CSS3 Animations (Keyframes) – Using Icons (Font Awesome / SVG) – Background Video – CSS Grid Basics – Parallax Scrolling Effect Basics

UNIT V DEPLOYMENT AND CASE STUDIES 9 hours

Planning & Designing a Simple Website – Creating Wireframes – Integrating HTML, CSS, and JavaScript – Interactive Navigation Menus – Form Validation & Error Handling – Optimizing Images & Media – Testing & Debugging – Web Hosting Options – Basics of Domain Names – Deploying a Simple Web Application – Website Maintenance & Updates.

Course Outcomes:

After completing this Unit, students will be able to

CO1: Demonstrate the ability to design structured and well-formatted web pages using HTML and HTML5 elements.

CO2: Apply CSS styling techniques to enhance the visual presentation and layout of web pages.

CO3: Develop interactive web pages with JavaScript for event handling, validation, and DOM manipulation.

CO4: Analyze design requirements to select and implement HTML5 and CSS3 features for responsive and accessible user interfaces.

CO5: Integrate and deploy a functional, responsive website by combining HTML, CSS, and JavaScript, followed by testing and debugging.

Text Book(s)

1. Adam Wathan and Steve Schoger, Refactoring UI: The Book for Better User Interface Design, 1st Edition, Self-Published, 2019.
2. Adham Dannaway, Practical UI: A Logic-Driven Approach to User Interface Design, 1st Edition, Self-Published, 2021.
3. Jenifer Tidwell, Designing Interfaces: Patterns for Effective Interaction Design, 3rd Edition, O'Reilly Media, 2019.

Reference Books

1. Alan Cooper, Robert Reimann, David Cronin, and Christopher Noessel, About Face: The Essentials of Interaction Design, 4th Edition, Wiley, 2014.
2. Don Norman, The Design of Everyday Things, Revised and Expanded Edition, Basic Books, 2013

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

B. Tech I Year I 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 I 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:

Students will be able to:

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 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

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

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 II 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 II 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 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

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 II 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.

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 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

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 I Semester

25BACOMTC01 COMPUTER ORGANIZATION AND ARCHITECTURE

L	T	P	C
3	0	0	3

Pre-requisite: Digital Design

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

25BACSETC04 DATABASE MANAGEMENT SYSTEMS

L	T	P	C
3	0	0	3

Pre-requisite: Data Structures and Algorithms

Course Description:

This course is designed to provide a basic understanding of database systems and their design. The course material is further used for developing any web-based applications in which the database is back end. The course covers all basic and advanced queries of SQL, PL/SQL programs, Relational algebra and calculus, normal forms, low-level details such as representing data elements of the database and indexed structures, transaction management, and data recovery.

Course Objectives:

1. To understand the concept of DBMS and ER Modelling.
2. To comprehend the structure of SQL Queries and commands to manage data from the databases
3. To learn PL/SQL concepts that help in seamless processing of SQL
4. To explain the normalization, Query optimization, and relational algebra
5. To apply Transaction processing, concurrency control, recovery, security, and indexing for the real-time data
6. To gain knowledge on Database Attacks, Recovery, and Recent Trends

UNIT I INTRODUCTION **9 hours**

Database Systems Concepts and Architecture: History and motivation for database systems-characteristics of database approach Advantages of using DBMS approach- Architectures for DBMS– Classification of database management systems.
Database Modelling: Types of Attributes, Entities, Relationships, ER Model.

UNIT II RELATIONAL MODEL **9 hours**

Relational Data Model: Concept of relations, schema-instance distinction, keys, referential integrity, foreign keys, relational algebra operators, SQL - Introduction, data definition in SQL, table, key, and foreign key definitions. Querying in SQL, notion of aggregation, Integrity constraints, aggregation functions group by and having clauses.
PL/SQL concepts: Triggers, Cursors, Introduction to JDBC, Stored Procedures.
Relational Algebra and Calculus: Preliminaries, Relational algebra- Selection and Projection, Set Operations, Renaming, Joins, Division. Relational Calculus.

UNIT III DATABASE DESIGN & SCHEMA REFINEMENT **9 hours**

Database Design: Dependencies and Normal forms, dependency theory - functional dependencies, Armstrong's axioms for FD's, closure of a set of FDs.
Normalization: 1NF, 2NF, 3NF, BCNF, 4NF, and 5NF decompositions and desirable properties.

UNIT IV TRANSACTION PROCESSING & INDEXING **9 hours**

Transaction processing - Concepts of transaction processing, ACID properties, concurrency control, Time-stamp based and lock-based protocols for concurrency control. Serializability of scheduling
Index Structures – Indexes on Sequential Files – Secondary Indexes – B-Trees – Hash Tables.

UNIT V NoSQL **9 hours**

Introduction to NoSQL databases, need and features of NoSQL, SQL vs NoSQL comparison, types of NoSQL databases: key-value, document, column-family and graph databases, basic MongoDB operations

Course Outcomes:

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

CO1: Apply design principles for database design, ER model
CO2: Demonstrate the basics of query evaluation and heuristic query optimization techniques
CO3: Access normalization relations of the relational model using normal forms
CO4: Implement transaction processing techniques in the database.
CO5: Understand the basics of NoSQL databases and recent database technologies.

Text Book(s)

1. Database Management Systems, Raghu RamaKrishnan, Johannes Gehrke, 3rd Edition, 2003, McGraw Hill.
2. Database Systems, The Complete Book, Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom, 3rd impression, 2009, Pearson.

Reference Books

1. Silberschatz,H.F.Korth S.Sudershan, Database System Concepts, McGraw Hill, 6th Ed., 2010.
2. R. Elmasri S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 2015
3. Thomas Connolly, Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation, and Management,6th Edition,2012.
4. Pramod J. Sadalage and Marin Fowler, NoSQL Distilled: A brief guide to merging world of Polyglot persistence, Addison Wesley, 2012

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

B. Tech II Year I Semester

25BACSETC05 OBJECT ORIENTED PROGRAMMING

L	T	P	C
3	0	0	3

Pre-requisite: C Programming for Problem Solving

Course Description:

Basics of Object-Oriented Programming - objects, classes, polymorphism, inheritance, static and dynamic binding. Object Oriented Programming using Java-classes, interfaces, inheritance, polymorphism, method dispatch, features for encapsulation and modularity.

Course Objectives:

1. Understand object-oriented programming concepts, and apply them in solving problems.
2. Learn the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes.
3. To Introduce the implementation of packages and interfaces.
4. Learn the concepts of exception handling and multithreading.
5. Learn the design of Graphical User Interface using applets and swing controls.

UNIT I INTRODUCTION TO OOPS CONCEPTS AND CLASSES 9 hours

Introduction to Object Oriented Programming, characteristics of Java, Java Programming Basics, Sample programs, Data types and operators, Control statements.

Classes: Classes, Objects, Methods, Constructors, this and static keywords, Method and Constructor Overloading, Access modifiers, Polymorphism

Arrays: One Dimensional and multi-dimensional arrays.

UNIT II STRINGS, INHERITANCE, INTERFACES, AND PACKAGES 9 hours

Strings: Strings, String Handling

Inheritance: Basics, Usage of Super, Multi-level hierarchy, Method overriding, Abstract class, Final keyword.

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

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

UNIT III EXCEPTION HANDLING &MULTI-THREADING 9 hours

Exception Handling: Fundamentals, Types, Multiple catch clauses, Nested try blocks, Thrown Class, Using Finally and Throws, Built-in exceptions, User-defined exceptions.

Multi-threading: Thread Class, Runnable interface, creating multiple threads, life cycle of thread, thread properties, synchronization, thread communication, suspending, resuming and stopping threads.

UNIT IV I/O STREAMS AND COLLECTION FRAME WORK CLASSES 9 hours

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

Collection Frame work : Hierarchy of collection framework, Array-List, Linked-List, Vector, Stack, Queue, Priority Queue, Hash Set, Linked Hash Set, Tree Set, Hash maps.

UNIT V SPRING MVC 9 hours

Introduction to Swing, basic Swing components, containers, event handling. Introduction to JDBC, database connection, executing SQL queries, retrieving results, and updating database records.

Introduction to Spring MVC, MVC architecture, controller, model, view, request mapping, form handling, validation, JDBC integration, and development of simple web applications.

Course Outcomes:

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

CO1: Choose object-oriented programming concepts for problem solving.
CO2: Create and use packages and interfaces.
CO3: Develop multithreaded applications with synchronization.
CO4: Provide computed based solutions by using java collection framework and I/O classes.
CO5: Understand the basics of Spring MVC and develop simple web applications.

Text Books:

1. Java The Complete Reference, Herbert Schildt, MC GRAW HILL Education, 9th Edition, 2016.

Reference Books

1. Core Java Volume I – Fundamentals, by Cay S. Horstmann, Gary Cornell Pearson Education Ninth Edition
2. "Java Fundamentals - A Comprehensive Introduction", Herbert Schildt and Dale Skrien, Special Indian Edition, McGrawHill, 2013.
3. "Java – How to Program", Paul Deitel, Harvey Deitel, PHI.
4. "Thinking in Java", Bruce Eckel, Pearson Education.
5. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.

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

B. Tech II Year I Semester

25BACSETC06 ANALYSIS OF ALGORITHMS

L	T	P	C
3	0	0	3

Pre-requisite: Data Structures and Algorithms

Course Description:

This course emphasis on analysis of various types of algorithms. It provides idea to design the algorithm to solve the problems using divide and conquer, greedy method, dynamic programming, backtracking, branch and bound, approximation.

Course Objectives:

1. To introduce the concepts of Algorithm Analysis, Time Complexity, Space Complexity.
2. To discuss various Algorithm Design Strategies with proper illustrative examples.
3. To introduce Complexity Theory with NP and Approximation.

UNIT I INTRODUCTION & DIVIDE AND CONQUER 9 hours

Introduction: What is an algorithm?, Algorithm specification, Space Complexity, Time Complexity, Orders of Growth, Worst-Case, Best-Case, and Average-Case Efficiencies, Asymptotic notations.

Divide and Conquer: Master's Method, Substitution Method, Recursion Tree Method, Binary Search, Finding the maximum and minimum, Merge sort, Quick Sort, Strassen's matrix multiplication.

UNIT II GREEDY METHOD & DYNAMIC PROGRAMMING 9 hours

Greedy Method: General method, Fractional Knapsack problem, Huffman Code, Job Scheduling with Deadlines, Optimal merge pattern.

Dynamic Programming: General method, String Editing, Longest Common Subsequence, Matrix Chain Multiplication, 0/1 Knapsack problems, The traveling sales person problem.

UNIT III GRAPH ALGORITHMS 9 hours

BFT, DFT, Connected components, Biconnected Components, Spanning Trees, Minimum cost Spanning Trees, Kruskal's and Prim's algorithm, Topological sort, Shortest Path Algorithms: Dijkstra's Single Source Shortest Path Algorithm, Floyd-Warshall's All Pairs Shortest Path Algorithm.

UNIT IV BACK TRACKING & BRANCH AND BOUND 9 hours

Backtracking: General method, N-Queens Problem, Sum of subset problem, Graph Coloring Problem.

Branch and Bound: General method: FIFO, LIFO and LC, Travelling salesperson problem, 0/1 Knapsack problem.

UNIT V NP PROBLEMS & APPROXIMATION ALGORITHMS 9 hours

NP Problems: Complexity Class - P, NP, NP Complete, NP Hard. Reducibility, Cook's Theorem.

Approximation Algorithms: Introduction, Absolute Approximation, ϵ - Approximation, Polynomial time Approximation.

Course Outcomes:

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

CO1: Analyze the performance of different algorithms.

CO2: Identify optimal solution for different problems using greedy method and dynamic programming.

CO3: Implement various graph based algorithms.

CO4: Make use of backtracking and branch & Bound methods to solve real world problems.

CO5: Understand the complexity of NP problems and Approximation algorithms.

Text Books:

1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2008
2. Jon Kleinberg and Eva Tardos "Algorithm Design", Pearson Education, 2007

Reference Books

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Third Edition, Pearson Education, 2012
2. Micheal T. Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet examples", Second Edition, Wiley Publication, 2006
3. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education, 2006

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

B. Tech II Year I Semester

25BACSELC05 DATABASE MANAGEMENT SYSTEMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Nil

Course Description:

This course is designed to provide a basic understanding of database systems and their design. The course material is further used for developing any web-based applications in which the Database is back end. The course covers all basic and advanced queries of SQL, PL/SQL programs, low-level details such as representing data elements of Databases.

Course Objectives:

1. To understand the concept of DBMS and ER Modelling.
2. To understand the components of DBMS and to study database design.
3. To comprehend the structure of SQL Queries and commands to manage data from the databases
4. To comprehend the structure of SQL Queries to query, update, and manage a database.
5. To understand all constraints to develop a business application using cursors, triggers, and stored procedures

List of Programs:

1. **Design ER Model for Student Course Registration System**
   Identify entities, attributes and relationships for students, departments, faculty, courses, course registration and marks. Draw the ER diagram.
2. **Create Common Database and Basic Tables**
   Create a database named `StudentCourseDB` and create tables such as Department, Student, Faculty, Course and Semester.
3. **Apply Keys and Integrity Constraints**
   Apply primary keys, foreign keys, unique constraints, NOT NULL constraints and check constraints for the created tables.
4. **Insert and Display Records**
   Insert sample records into Department, Student, Faculty and Course tables and display the records using basic SELECT queries.
5. **Implement Course Registration Module**
   Create a CourseRegistration table to register students for courses and link it with Student and Course tables using foreign keys.
6. **Perform SQL Queries for Student and Course Details**
   Write SQL queries to retrieve student details, course details, department-wise students and courses registered by each student.
7. **Use Aggregate Functions and Grouping**
   Use COUNT, AVG, MIN, MAX and SUM to find number of students in each department, number of students registered for each course and average marks.
8. **Implement Joins for Report Generation**
   Use inner join, left join and right join to generate reports such as student-course list, faculty-course list and department-wise student list.
9. **Implement PL/SQL Procedures, Cursors and Triggers**
   Create a stored procedure to display registered courses of a student, use cursor to process marks, and create trigger to validate mark entry.

10. **Apply Normalization Techniques**
    Take an unnormalized student-course-registration table and convert it into **1NF, 2NF, 3NF, BCNF, 4NF and 5NF** by identifying functional dependencies and removing redundancy.
11. **Implement Transaction Processing and Indexing**
    Demonstrate COMMIT, ROLLBACK and SAVEPOINT during course registration and mark entry. Create indexes on student roll number, course code and department ID.
12. **Implement NoSQL using MongoDB for Student Feedback**
    Create a MongoDB collection to store student feedback for courses. Perform insert, update, delete and search operations using document-based data.

Course Outcomes:

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

CO1: Perform DDL and DML operations on database tables.
CO2: Design and implement complex queries to access the data using SQL join.
CO3: Implement stored procedures in PL/SQL.
CO4: Implement exceptions and triggers to solve real-time problems.
CO5: Design and develop a real-world application to access and render data.

Text Books:

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

Reference Books

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

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

B. Tech II Year I Semester

25BACSELC06 OBJECT ORIENTED PROGRAMMING LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: C Programming for Problem Solving

Course Description:

Basics of Object-Oriented Programming - objects, classes, polymorphism, inheritance, static and dynamic binding. Object Oriented Programming using Java-classes, interfaces, inheritance, polymorphism, method dispatch, features for encapsulation and modularity.

Course Objectives:

1. Understand object-oriented programming concepts, and apply them in solving problems.
2. Learn the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
3. To Introduce the implementation of packages and interfaces
4. Learn the concepts of exception handling and multithreading.
5. Learn the design of Graphical User Interface using applets and swing controls

List of Programs:

1. a) Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative,display a message stating that there are no real solutions.
   b) Write a Java program that find prime numbers between 1 to n.
2. a) The Fibonacci sequence is defined by the following rule: The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that print the nth value in the Fibonacci sequence.
   b) Develop a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a Palindrome
3. a) Implement a java program to split a given text file into n parts. Name each part as the name ofthe original file followed by part<n> where n is the sequence number of the part file
   b) Implement a java program to convert an Array List to an Array.
4. a) Develop a a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
   b) Develop a a Java program that reads a file and displays the file on the screen, with a line number before each line.
5. a) Implement java program to create a super class called Figure that receives the dimensions oftwo dimensional objects. It also defines a method called area that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and second is Triangle. Each of the sub class overridden area() so that it returns the area of a rectangle and a triangle respectively.
   b) Develop a a Java program that creates three threads. First thread displays ―Good Morning‖every one second, the second thread displays ―Hello‖ every two seconds and the third thread displays ―Welcome‖ every three seconds
6. Use inheritance to create an exception super class called Exception A and exception sub class Exception B and Exception C, where Exception B inherits from Exception A and

Exception C inherits from Exception B. Write a java program to demonstrate that the catch block for type Exception A catches exception of type Exception B and Exception C

7. Develop an application for simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result
8. Design & Develop an application that creates a user interface to perform integer divisions. The user enters two numbers in the Jtext Fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.
9. a) Design a GUI application for Cafeteria bill generation.
   b) Create a database connection using JDBC & perform some basic operation such as add, remove, update record in database using JDBC.

Course Outcomes:

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

CO1: Solve real world problems using OOP techniques.

CO2: Implement string handling and file handling methods.

CO3: Design multithreaded applications with synchronization.

CO4: Develop web applications using AWT components.

CO5: Create GUI based applications

Text Books:

1. Java The Complete Reference, Herbert Schildt, MC GRAW HILL Education, 9thEdition, 2016.

Reference Books

1. Core Java Volume I – Fundamentals, by Cay S. Horstmann, Gary Cornell Pearson Education Ninth Edition
2. "Java Fundamentals - A Comprehensive Introduction", Herbert Schildt and Dale Skrien, Special Indian Edition, McGrawHill, 2013.
3. "Java – How to Program", Paul Deitel, Harvey Deitel, PHI.
4. "Thinking in Java", Bruce Eckel, Pearson Education.
5. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.

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

B. Tech II Year I Semester

25BACSELC07 ANALYSIS OF ALGORITHMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Data Structures and Algorithms

Course Description:
This course is aimed to provide hands on experience to analyse the time complexity of sorting, graph based, greedy, dynamic programming and backtracking algorithms.

Course Objectives:
1. To learn basic algorithm design methods.
2. To understand time and space complexity of algorithms.
3. To solve real-life problems using suitable algorithms.
4. To implement searching, graph, and optimization algorithms.
5. To understand exact and approximate solutions for difficult problems.

List of Programs:

1. A library stores book IDs in sorted order. Implement a system to search a book ID quickly.
2. An examination cell wants to find the highest and lowest marks from a large student dataset.
3. A messaging application wants to compress messages using shorter codes for frequently used characters.
4. A plagiarism detection system compares two documents and finds the longest common sequence.
5. A telecom company wants to connect all cities with minimum cable cost.
6. A map application wants to find the shortest path from one city to all other cities.
7. Select a subset of student grants so that the total amount matches a fixed scholarship fund.
8. A delivery company wants to find the minimum-cost route to visit all cities and return to the start.
9. A logistics company needs a near-optimal route when finding the exact TSP solution is costly.
10. A company wants to select resources within a budget and achieve near-optimal value efficiently.

Course Outcomes:
Upon successful completion of the course, students will be able to
CO1: Write programs using different algorithm design techniques.
CO2: Analyze the efficiency of algorithms.
CO3: Apply algorithms to solve searching, compression, comparison, and routing problems.
CO4: Solve optimization problems using greedy, dynamic programming, backtracking, and branch and bound methods.
CO5: Use approximation algorithms when exact solutions are difficult or time-consuming.

Text Books:
1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Second Edition, Universities Press, 2008
2. Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Third Edition, Pearson Education, 2012.

Reference Books:

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited, 2012.
2. S. Sridhar, "Design and Analysis of Algorithms", Oxford university press, 2014.
3. Web reference: http://nptel.ac.in/

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

II YEAR II SEMESTER

B. Tech II Year II Semester

25BAMATTC09 PROBABILITY AND STATISTICS FOR COMPUTER SCIENCE

L	T	P	C
3	0	0	3

Pre-requisite 25BBMATTC01

Course Description:

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

Course Objectives:

The objectives of this course are:

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

UNIT I PROBABILITY AND RANDOM VARIABLES 9 hours

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

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

UNIT II PROBABILITY DISTRIBUTIONS AND APPLICATIONS 9 hours

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

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

UNIT III JOINT DISTRIBUTIONS AND DEPENDENCY ANALYSIS 9 hours

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

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

UNIT IV STATISTICAL DATA ANALYSIS AND PREDICTIVE MODELLING 9 hours

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

UNIT V HYPOTHESIS TESTING AND STATISTICAL DECISION MAKING 9 hours

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

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

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

Course Outcomes:

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

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

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

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

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

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

Text Books

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

Reference Books

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

E Books

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

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

B. Tech II Year II Semester

25BACSETC07 MACHINE LEARNING AND APPLICATIONS

L	T	P	C
3	0	0	3

Pre-requisite: Object Oriented Programming

Course Description:

This course provides a comprehensive introduction to machine learning, covering both foundational concepts and advanced techniques. Students will explore supervised, unsupervised, and reinforcement learning, along with model evaluation and deployment methods. Emphasis is placed on practical implementation using real-world datasets and case studies.

Course Objectives:

This course enables students to

1. To introduce the fundamental concepts and types of machine learning.
2. To develop a deep understanding of supervised and unsupervised learning algorithms.
3. To understand mathematical foundations of learning models and algorithms.
4. To evaluate model performance using appropriate statistical and analytical tools.

UNIT I INTRODUCTION TO MACHINE LEARNING 9 hours

Definition and Scope of Machine Learning, Applications and Types of Learning: Supervised, Unsupervised, Reinforcement, Linear Regression: Least Squares, Cost Function, Gradient Descent, Polynomial Regression and Overfitting, Evaluation Metrics: RMSE, MAE, R^2 Score, Bias-Variance Trade off.

UNIT II CLASSIFICATION ALGORITHMS 9 hours

Classification Overview and Decision Boundaries, Logistic Regression: Sigmoid Function and Cost, K-Nearest Neighbors (KNN), Naïve Bayes Classifier, Decision Trees and Random Forests, Model Evaluation: Confusion Matrix, Precision, Recall, F1-Score.

UNIT III SUPPORT VECTOR MACHINES AND ENSEMBLE METHODS 9 hours

Support Vector Machines: Concepts, Kernels, Hyperplane and Margin Concepts, Kernel Tricks: RBF and Polynomial, Ensemble Learning: Bagging, Boosting, and Voting, Gradient Boosting, AdaBoost, and XGBoost, Model Tuning and Hyperparameter Optimization.

UNIT IV UNSUPERVISED LEARNING TECHNIQUES 9 hours

Clustering Overview: Applications, K-Means Clustering Algorithm, Hierarchical Clustering, DBSCAN and Density-Based Methods, Principal Component Analysis (PCA) for Dimensionality Reduction, Silhouette Score, Davies-Bouldin Index for Cluster Validation.

UNIT V ADVANCED TOPICS AND APPLICATIONS 9 hours

Reinforcement Learning Basics and Markov Decision Processes, Cross-Validation Techniques: k-Fold, Leave-One-Out, Feature Engineering and Feature Selection, Deployment of ML Models (Flask, Streamlit, etc.), Case Studies: Medical Diagnosis, Spam Detection, Credit Scoring.

Course Outcomes:

After completing this Unit, students will be able to

CO1: Outline the foundational structure of machine learning systems and linear regression techniques.

CO2: Experiment with different classification models and evaluate their performance using standard metrics.

CO3: Apply support vector machines and ensemble learning techniques to classification problems.

CO4: Discover meaningful clusters in datasets using unsupervised learning and dimensionality reduction.

CO5: Analyze advanced machine learning techniques and their application through real-world case studies.

Text Book(s)

1. Tom M. Mitchell, Machine Learning, McGraw-Hill Education, 1st Edition, 1997, ISBN: 978-0070428072.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 2nd Edition, 2019, ISBN: 978-1492032649.
3. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 4th Edition, 2020, ISBN: 978-0262043793.

Reference Books

1. Trevor Hastie, Robert Tibshirani & Jerome Friedman, The Elements of Statistical Learning, Springer, 2nd Edition, 2009, ISBN: 978-0387848570.
2. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 1st Edition, 2012, ISBN: 978-0262018029.
3. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 1st Edition, 2006, ISBN: 978-0387310732.

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

B. Tech II Year II Semester

25BACSETC08 DATA COMMUNICATION AND COMPUTER NETWORKS

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Description:
The main emphasis of this course is to understand the basic concepts of Computer Networks, Introducing the layered approach for design of computer networks, introduce to computer communication, TCP/IP layers functionalities, and operations of network protocols in the TCP/IP suite, and elucidating the design issues for a computer network.

Course Objectives:
1. To study the evolution of computer networks, foundational principles, architectures, and techniques employed in computer networks.
2. To study the concepts of communication networks from layered perspective
3. To provide students with a theoretical and practical base in computer networks issues
4. Student will be able purse his study in advanced networking courses
5. To Prepare students for easy transfer from academia into future directions of research.

UNIT I INTRODUCTION **9 hours**

NETWORK FUNDAMENTALS:
Introduction, Advantages and Applications, Network Types, Topologies, Internet History, Standards and Administration. **Network Models:** Protocol Layering, The ISO Model, TCP/IP Protocol Suite, Cross-layering.

THE PHYSICAL LAYER
Data and Signals, Transmission impairment, Data rate limits, Performance. **Transmission media:** Introduction, Guided Media, Unguided Media. **switching:** Circuit Switched Networks, Packet switching.

UNIT II THE DATA LINK LAYER **9 hours**
Introduction, Link layer addressing. **Error detection and Correction:** Cyclic codes, Checksum, Forward error correction. **Data link control:** DLC Services, Data link layer protocols, HDLC, Point to Point Protocol. **Media Access control:** Random Access, Controlled Access, Channelization, and connecting devices.

UNIT III THE NETWORK LAYER **9 hours**
Network layer design issues, Routing algorithms, (Optimal, Shortest path, Distance Vector routing, Link State routing, Hierarchical routing, Routing in adhoc networks), Congestion control algorithms, Quality of service, Internetworking, **The network layer in the Internet:** IP Classes, IPV4 and IPV6, IP Addressing, NAT, IP support protocols, OSPF, BGP, RIP, IGMP.

UNIT IV THE TRANSPORT LAYER **9 hours**
The Transport Service, Elements of Transport Protocols, Flow control, Congestion Control, **The internet transport protocols:** UDP, TCP, SCTP. Performance problems in computer networks, Network performance measurement.

UNIT V THE APPLICATION LAYER 9 hours

Functions of session layer, presentation layer and application layer, client-server programming, WWW and HTTP, FTP, e-mail, TELNET, Secure Shell, Domain Name System and SNMP.

Course Outcomes:

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

CO1: Understand and choose the transmission media and topologies depending on the requirements.

CO2: Apply error detection and error correction wherever required.

CO3: Analyze the concepts of routing, and congestion control

CO4: Evaluate the computer network logically, by enumerating the layers of the TCP/IP.

CO5: Create and make use of application-level protocols for file communication, and file transfer.

Text Book(s)

1. "Data communications and networking", Behrouz A. Forouzan, Mc Graw Hill Education,5th edition, 2012.
2. "Computer Networks", Andrew S. Tanenbaum, Wetherall, Pearson,6th edition, 2021.

Reference Books

1. "Internetworking with TCP/IP – Principles, protocols, and architecture", Volume 1, Douglas E. Comer, 5th edition, PHI
2. Peterson, Larry L., and Bruce S. Davie. Computer networks: a systems approach. Elsevier, 2007.
3. "Data communications & networking with TCP/IP protocol suite", Behrouz A. Forouzan, Mc Graw Hill Education, 2021.
4. Droms, R. (2001). Computer networks and internets: with internet applications. Prentice Hall.

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

B. Tech II Year II Semester

25BACSETC09 OPERATING SYSTEMS

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 describe the operating system structures, services and system call.
2. To Demonstrate process management concepts and threads
3. To illustrate scheduling algorithms and process synchronization and deadlocks.
4. To analyze various memory management techniques.
5. To understanding of file system concepts and implementations.

UNIT I INTRODUCTION 9 hours
Introduction to Operating Systems: Introduction, Operating system functions, Operating systems operations, **System Structures:** Operating System Services & Interrupts, system calls, Kernel approaches, Building and booting an OS, debugging

UNIT II PROCESS CONCEPTS 9 hours
Processes: Process Concept, Process scheduling, Operations on processes, Inter-process communication. **Threads and Concurrency:** Multithreading models, Thread libraries, Threading issues. **CPU Scheduling:** Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling

UNIT III PROCESS SYNCHRONIZATION & DEADLOCK 9 hours
Process Synchronization & Deadlocks: The Critical Section Problem, Peterson's Solution, Semaphores, Monitors, Classic problems of Synchronization. **Deadlocks:** System Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock

UNIT IV MEMORY MANAGEMENT 9 hours
Memory-Management Strategies: Introduction, Contiguous memory allocation, Paging, Structure of the Page Table, Swapping. **Virtual Memory Management:** Introduction, Demand paging, Page replacement, Allocation of frames, Thrashing. **Storage Management:** Overview of Mass Storage Structure, HDD Scheduling.

UNIT V FILE SYSTEMS 9 hours
File System: File concept - Access methods - Directory Structure - Directory organization - File system mounting - File Sharing and Protection, File System Implementation - File System Structure – Directory implementation - Allocation Methods - Free Space Management.

Case Studies: Linux System- Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management, Input-Output Management, File System, Inter-Process Communication, Virtual Memory Management.

Course Outcomes:

At the completion of the course the students will be able to:
CO1: Understand the functions & design principles of operating systems.
CO2: Apply process scheduling algorithms and synchronization techniques to enhance computer system performance.
CO3: Analyze the conditions for deadlock and the effectiveness of possible solutions.
CO4: Evaluate different memory management techniques and allocation policies in operating systems.
CO5: Design files allocation strategies to secure data in operating systems.

Text Books:

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

Reference Books:

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

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

B. Tech II Year II Semester

25BACSELC08 MACHINE LEARNING AND APPLICATIONS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Object Oriented Programming

Cour₹se Description:

This lab course offers practical exposure to foundational and advanced machine learning algorithms through hands-on programming in Java or Python. Students will implement, test, and evaluate models such as decision trees, neural networks, Bayesian classifiers, and clustering techniques. The course emphasizes real-world applications and project-based learning using standard datasets.

Course Objectives:

This course enables students to

1. To introduce students to basic machine learning techniques using Python programming.
2. To train students to implement regression and classification algorithms for real-world prediction problems.
3. To enable students to evaluate machine learning models using suitable performance metrics.
4. To provide practical knowledge of clustering, ensemble learning, and support vector machine algorithms.
5. To develop the ability to build and deploy simple machine learning applications using Flask or Streamlit.

List of Experiments

Note:

a. The programs can be implemented in either JAVA or Python.

b. Data sets can be taken from standard repositories (https://archive.ics.uci.edu/ml/datasets.html)/kaggle/ constructed by the students/any open source repositories.

1. Implement simple linear regression for predicting house price based on area and evaluate the model using MAE, RMSE and R^2 score.
2. Implement linear regression using gradient descent for predicting salary based on years of experience.
3. Implement logistic regression to predict student placement based on CGPA and aptitude score.
4. Apply K-Nearest Neighbors algorithm to classify patients as diabetic or non-diabetic using medical data.
5. Implement Naïve Bayes classifier for spam email detection and evaluate using confusion matrix, precision, recall and F1-score.
6. Develop a decision tree classifier for loan approval prediction based on customer income, credit score and employment status.
7. Apply Random Forest algorithm for credit risk classification and compare its performance with decision tree.
8. Implement Support Vector Machine for tumor classification using linear, polynomial and RBF kernels.
9. Apply ensemble learning methods such as Voting, AdaBoost or Gradient Boosting for customer churn prediction.

10. Implement K-Means clustering for customer segmentation based on income and spending score.
11. Apply hierarchical clustering for grouping customers based on purchasing behavior and visualize the dendrogram.
12. Develop and deploy a simple machine learning prediction application using Flask or Streamlit.

Course Outcomes:

After completing this Unit, students will be able to

CO1: Implement regression models for prediction problems and evaluate them using suitable metrics.

CO2: Apply classification algorithms such as logistic regression, KNN, Naïve Bayes, decision tree and random forest.

CO3: Use SVM and ensemble learning methods to solve classification problems

CO4: Apply clustering techniques such as K-Means and hierarchical clustering for grouping data.

CO5: Develop and deploy a simple machine learning application using Flask or Streamlit

Text Book(s)

1. Python Machine Learning Workbook for Beginners, AI Publishing, 1st Edition, 2020, ISBN: 978-1734790160.

Online Learning Resources/ Virtual Labs

1. Kirill Eremenko & Hadelin de Ponteves, Machine Learning A-Z™: Hands-On Python & R In Data Science, Udemy,OnlineCourse, https://www.udemy.com/course/machinelearning/
2. Andrew Ng, Machine Learning, Coursera, Stanford University, Online Course, https://www.coursera.org/learn/machine-learning

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

B. Tech II Year II Semester

25BACSELC09 DATA COMMUNICATION AND COMPUTER NETWORKS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite Nil

Course Description:

This lab course focuses on practical aspects of computer networks, including graph algorithms, TCP/UDP client-server programming, multiple client communication, error detection, routing protocols, and network configuration. Students gain hands-on experience in simulation, implementation, and performance analysis of reliable data communication.

Course Objectives:

1. To understand the basic working of computer networks through simulation.
2. To simulate data transmission, encoding, framing, and error detection techniques.
3. To study flow control and reliable data transfer using sliding window protocols.
4. To analyze packet transfer, packet loss, congestion, and throughput using NS simulator.
5. To observe and analyze real-time network packets and protocols using Wireshark.

List of Experiments:

1. A client sends a message to a server. Simulate the working of OSI layers and show how data is encapsulated at the sender side and decapsulated at the receiver side.
2. A digital communication system transmits binary data. Simulate NRZ, NRZ-I, and Manchester encoding and compare the signal representation for the same input data.
3. A data link layer system needs to divide continuous data into frames. Simulate character count, bit stuffing, and byte stuffing methods and recover the original message.
4. A sender transmits frames to a receiver over an unreliable channel. Simulate Go-Back-N ARQ and Selective Repeat ARQ to handle lost or damaged frames.
5. A receiver wants to verify whether the received data contains errors. Simulate CRC polynomial-based error detection and display the transmitted and received CRC values.
6. Two computers are connected through a simple network. Simulate data transfer between them using NS and observe packet flow, delay, and delivery status.
7. A network link becomes overloaded during communication. Simulate packet loss using NS by changing queue size or bandwidth and identify the dropped packets.
8. A TCP connection experiences congestion while transferring data. Simulate congestion control using NS and observe changes in sending rate and congestion window.
9. A small network has multiple point-to-point links. Simulate the network by varying bandwidth and queue size, then compare packet drops under different conditions.
10. A laboratory LAN generates different types of traffic. Simulate or capture network packets using NS/Wireshark and analyze throughput, protocol type, and packet count.

Course Outcomes:

After completion of the course, students will be able to

CO1: Simulate OSI layer-based communication between client and server.

CO2: Implement encoding, framing, and CRC-based error detection techniques.

CO3: Demonstrate reliable data transfer using Go-Back-N ARQ and Selective Repeat ARQ.

CO4: Analyze network performance by measuring packet loss, throughput, congestion, and dropped packets.

CO5: Use NS simulator and Wireshark for network simulation, packet capture, and protocol analysis.

Text Books:

1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, UNIX Network Programming, Volume 1: The Sockets Networking API, 3rd Edition, Pearson, 2004.
2. W. Richard Stevens, UNIX Network Programming, Volume 2: Interprocess Communications, 2nd Edition, Pearson, 1999.

References Books:

1. Andrew S.Tanenbaum, David j.wetherall, Computer Networks, 6th Edition, Pearson.
2. James F.Kurose, Keith W. Ross, Computer Networking: A Top-Down 6th edition, Pearson, 2019.
3. Computer Networks: A Systems Approach-Bruce Davie, VMware-Larry Peterson, Princeton University-2019.

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

B. Tech II Year II Semester

25BACSELC10 OPERATING SYSTEMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite: Nil

Course Description:

This course is designed to provide a basic understanding of database systems and their design. The course material is further used for developing any web-based applications in which the Database is back end. The course covers all basic and advanced queries of SQL, PL/SQL programs, low-level details such as representing data elements of Databases.

Course Objectives:

1. To implement various CPU scheduling algorithms.
2. To implement Deadlock Avoidance and Detection Algorithms
3. To implement Page Replacement Algorithms
4. To implement various memory allocation File Allocation methods.

List of Programs:

1. Demonstrate a program on System Calls
    - Extend the program to perform file operations using system calls:
        - open() – Open or create a file
        - write() – Write data to the file
        - read() – Read data from the file
        - close() – Close the file
    - Display the contents read from the file on the console.
2. Simulate the following CPU scheduling algorithms using suitable data structures and calculate waiting time and turnaround time:
    - FCFS
    - SJF
    - Round Robin
3. Implement process synchronization using semaphores and Mutex to prevent race conditions while applying in the Producer–Consumer problem.
4. Generate a set of memory partitions and process memory requirements, simulate the following allocation strategies and show the allocation result:
    - First Fit
    - Best Fit
    - Worst Fit
5. Simulate the page replacement algorithms FIFO, LRU, and Optimal for the given reference string. Calculate the total number of page faults for each algorithm.
6. Implement the Banker's Algorithm to detect and avoid deadlocks in a system with multiple processes and resources. Show safe state and unsafe state in sequence of process execution.
7. Implement Dining Philosopher Problem using semaphore to prevent dead lock, as when there are limited resources.
8. Simulate the following file allocation strategies and show how files are stored in memory:
    - Sequential Allocation
    - Linked Allocation
9. Implement the following disk scheduling algorithms and Estimate the total seek time for a given request queue.
    - FCFS (First-Come, First-Served)
    - SSTF (Shortest Seek Time First)

Course Outcomes:

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

CO1: Choose an appropriate Data Structure to perform CPU Scheduling & Job Scheduling algorithms
CO2: Implement process creation, termination, and scheduling algorithms for effective utilization of CPU.
CO3: Design solutions for synchronization problems using semaphores and mutexes to solve race conditions and deadlocks.
CO4: Compare the performance of memory management and page replacement algorithms.
CO5: Implement effective memory utilization using file allocation and disk scheduling algorithms

Text Books:

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

Reference Books

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

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

SKILL ENHANCEMENT COURSES

Skill Enhancement Course - I

25BACSESC01 DATA SCIENCE USING PYTHON

L	T	P	C
1	0	2	2

Pre-requisite: C Programming for Problem Solving**,** Data Structures and Algorithms

Course Objectives:

1. To learn basic Python programming concepts.
2. To use Python data structures for solving simple problems.
3. To write programs using functions, files, and exception handling.
4. To understand basic data science concepts using NumPy and Pandas.
5. To analyze and visualize data using Python libraries.

UNIT I INTRODUCTION TO PYTHON PROGRAMMING 6 hours

Introduction to Python - Features and applications of Python - Installing Python and using IDEs - Python syntax and indentation - Variables and data types - Input and output statements - Operators in Python - Conditional statements - Looping statements - Basic problem solving using Python

1. Develop a Python program to read student details such as name, roll number, branch, and marks, and display the complete student profile in a formatted manner.
2. Implement a Python program to calculate the electricity bill for a customer based on the number of units consumed using conditional statements.
3. Develop a Python program to read marks of a student and display the corresponding grade using if-else conditions.
4. Write a Python program to analyze a given number and check whether it is even or odd, positive or negative, and prime or not.

UNIT II PYTHON DATA STRUCTURES 6 hours

Strings and string operations - Lists and list operations - Tuples - Sets - Dictionaries - List comprehension.

1. Write a Python program to store marks of students in a list and find the highest mark, lowest mark, average mark, and sorted list of marks.
2. Write a Python program to read a sentence and count the number of vowels, consonants, words, and characters using string operations.
3. Write a Python program to remove duplicate student ID numbers from a list using set operations.
4. Write a Python program to store student names and marks in a dictionary and display the students who scored above a given mark.

UNIT III FUNCTIONS & FILE HANDLING 6 hours

Functions in Python - Function arguments - Lambda functions - Recursion - Built-in functions - File handling in Python - Reading and writing text files - Exception handling - Modules and packages

1. Write a Python program to design a simple calculator using functions to perform addition, subtraction, multiplication, and division.
2. Write a Python program to find the factorial of a number and generate the Fibonacci series using recursion.
3. Write a Python program to write student details into a text file and read the stored details from the file.
4. Write a Python program to perform division of two numbers using exception handling to manage division by zero and invalid input errors.

UNIT IV DATA SCIENCE FOUNDATIONS AND DATA HANDLING 6 hours

Introduction to Data Science - Data science lifecycle - Introduction to NumPy - NumPy arrays and operations - Introduction to Pandas - Series and DataFrame - Reading datasets using Pandas - Data cleaning - Handling missing values

1. Write a Python program to create a NumPy array of student marks and calculate the total, average, maximum, and minimum marks.
2. Write a Python program to create a Pandas DataFrame for product sales and calculate the total sales for each product.
3. Write a Python program to read a CSV dataset using Pandas and display the first five records, column names, and shape of the dataset.
4. Write a Python program to identify missing values in a dataset and replace them using mean, median, or default values.

UNIT V DATA ANALYSIS AND VISUALIZATION 6 hours

Data filtering and grouping - Data summarization using Pandas - Introduction to Matplotlib - Line chart - Bar chart - Pie chart - Histogram - Scatter plot - Basic data visualization for real-world datasets

1. Write a Python program to group student result data department-wise and calculate the average marks using Pandas.
2. Write a Python program to visualize monthly sales data using line chart and bar chart with Matplotlib.
3. Write a Python program to represent student grade distribution using pie chart and histogram.
4. Write a Python program to analyze weather data and display a scatter plot showing temperature variation over different days.

Course Outcomes:

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

CO1: Write simple Python programs using conditions and loops.
CO2: Use strings, lists, tuples, sets, and dictionaries in Python programs.
CO3: Create programs using functions, file handling, and error handling.
CO4: Perform basic data analysis using NumPy and Pandas.
CO5: Create simple charts and graphs using Matplotlib.

Text Books:

1. Gowri shankar S, Veena A., Introduction to Python Programming, CRC Press.
2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2nd Edition, Pearson, 2024.
3. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, O’Reilly, 2017.

Reference Books:

1. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.
2. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021.

Online Resources:

1. https://www.coursera.org/learn/python-for-applied-data-science-ai
2. https://www.coursera.org/learn/python?specialization=python#syllabus

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

Skill Enhancement Course - II

25BACSESC02 FULL STACK DEVELOPMENT

L	T	P	C
1	0	2	2

Pre-requisite : Fundamentals of Web Technology

Course Description:

This course introduces the fundamental concepts of full stack web development with emphasis on building interactive and data-driven web applications. It covers client-side development using JavaScript and React, server-side programming using Node.js and Express, and database management using MongoDB. The course focuses on integrating front-end and back-end components through RESTful APIs and performing basic CRUD operations. It also emphasizes scenario-based problem solving, enabling students to design and implement simple full stack applications for real-world web development challenges.

Course Objectives:

This course enables students to

1. To Understand fundamentals of Full Stack Web Development
2. To Build simple front-end applications using React
3. To Develop basic backend services using Node.js and Express
4. To Perform CRUD operations using MongoDB
5. To Integrate front-end and back-end in a simple full stack application

UNIT I JAVASCRIPT AND BASICS OF MERN STACK 9 hours

JavaScript basics: variables, functions, objects - DOM basics and event handling - Forms and basic validation - Introduction to MERN Stack - Overview of Node.js, React, Express, MongoDB - Installing Node.js and npm - Running basic JavaScript programs using Node

1) A college website stores student marks entered by a faculty member.
   Develop a JavaScript program to store student details, calculate total and average marks, an display the result on the webpage.
2) A student maintains a daily task list on a webpage.
   Implement JavaScript to add and delete tasks dynamically using event handling.
3) An online workshop registration page requires basic validation.
   Design a form using JavaScript that validates name, email, and mobile number.
4) A developer wants to verify Node.js installation.
   Write and execute a simple “Hello World” program using Node.js.
5) A student wants to understand the components of MERN stack.
   Identify the role of each MERN component using a simple example.

UNIT II REACT 9 hours

Introduction to React - JSX and components - Props and state - Event handling - Forms in React - Basic styling using CSS

1) An educational website displays a welcome message to logged-in users.
   Create a React component that displays a welcome message using JSX.
2) A student portal shows profile information.
   Create reusable React components and pass student data using props.
3) A teacher records student attendance using a web app.
   Implement a React component that increases or decreases attendance count using state and events.
4) A feedback page collects user responses.
   Design a React form to accept feedback and display submitted values.
5) A dashboard needs basic styling.
   Apply CSS styling to React components for better UI appearance.

UNIT III NODE.JS AND EXPRESS 9 hours

Node.js basics - Modules and npm - Express framework - Routing - REST API basics - HTTP methods - Middleware (basic idea)

1) A college wants to host a static webpage using Node.js.
   Develop a Node.js server to serve a static HTML page.
2) A library system wants to list available books.
   Create an Express API that returns book details in JSON format.
3) An online portal allows students to register.
   Implement a POST API using Express to accept student details.
4) A user submits incomplete data.
   Implement basic error handling in an Express application.
5) A developer wants to verify API functionality.
   Test GET and POST APIs using Postman.

UNIT IV MONGODB 9 hours

MongoDB introduction - Documents and collections - CRUD operations - Connecting MongoDB with Node.js - Basic aggregation - Backup and restore concepts

1) A company stores employee details.
   Create a MongoDB collection and insert employee records.
2) A department wants to manage student information.
   Perform CRUD operations on student documents.
3) A backend system stores user data.
   Connect MongoDB with Node.js and perform read and write operations.
4) An online store wants to know product count per category.
   Use MongoDB aggregation to generate a summary report.
5) An admin wants to secure data.
   Demonstrate MongoDB backup and restore procedure.

UNIT V ADVANCED FEATURES 9 hours

React Router (basic) - Connecting React with Express APIs - CRUD UI integration - Pagination and search (basic) - Session concept (overview only)

1) A website contains Home, About, and Contact pages.
   Implement client-side routing using React Router.
2) A college system manages student records.
   Integrate React UI with Express APIs to perform CRUD operations.
3) A dashboard displays data in table format.
   Design responsive UI components using Bootstrap.
4) A product list contains many items.
   Implement pagination and search functionality using MongoDB and React.
5) A user logs into a system.
   Explain session handling and demonstrate a simple login flow.

Course Outcomes:

After completing this course, students will be able to

CO1: Design basic client-side web pages using JavaScript and React components to handle user interaction and data presentation.

CO2: Develop server-side applications using Node.js and Express to process requests and generate appropriate responses.

CO3: Apply database concepts using MongoDB to perform Create, Read, Update, and Delete (CRUD) operations for real-world applications.

CO4: Integrate front-end, back-end, and database components to develop simple full stack web applications based on given problem scenarios..

CO5: Analyze application requirements and select appropriate full stack technologies to implement scenario-based web solutions effectively.

Text Book(s)

1. Shama Hoque, "Full-Stack React Projects: Learn MERN Stack Development by Building Modern Web Applications", Packt Publishing, 2nd Edition, 2020.
2. Vasan Subramanian, "Pro MERN Stack: Full Stack Web App Development with MongoDB, Express, React, and Node", Apress, 2nd Edition, 2019.
3. Greg Lim, "Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, and Node.js App", Independently Published, 1st Edition, 2018..

Reference Books

1. Brad Dayley, "Node.js, MongoDB and Angular Web Development", Addison-Wesley Professional, 2nd Edition, 2017.
2. Azat Mardan, "Practical Node.js: Building Real-World Scalable Web Apps", Apress, 2nd Edition, 2018.
3. Alex Banks and Eve Porcello, "Learning React: Functional Web Development with React and Redux", O'Reilly Media, 2nd Edition, 2020.

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