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

Estd 1998

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(CYBER SECURITY)
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
(CYBER SECURITY)

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(Deemed to be University)

MADANAPALLE

B. Tech Four Year Curriculum Structure

Branch: COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

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	25BACCSTC01	Fundamentals of Cyber Security	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	25BACCSTC02	Database Concepts and Applications	3	0	0	3	3
4	PCC	25BACCSTC03	Modern Operating Systems	3	0	0	3	3
5	PCC	25BACCSTC04	Design and Analysis of Computer Algorithms	3	0	0	3	3
6	PCC	25BACCSLC01	Database Concepts and Applications Laboratory	0	0	2	2	1
7	PCC	25BACCSLC02	Modern Operating Systems Laboratory	0	0	2	2	1
8	PCC	25BACCSLC03	Design and Analysis of Computer 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**	**24**	**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	25BACCSTC05	Principles of Object Oriented Programming	3	0	0	3	3
3	PCC	25BACCSTC06	Computer Networks	3	0	0	3	3
4	PCC	25BACCSTC07	Cryptography	3	0	0	3	3
5	OE		**Open Elective – I** *(Refer ANNEXURE - II)*	3	0	0	3	3
6	PCC	25BACCSLC04	Principles of Object Oriented Programming Laboratory	0	0	2	2	1
7	PCC	25BACCSLC05	Computer Networks Laboratory	0	0	2	2	1
8	PCC	25BACCSLC06	Cryptography 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		Financial Technology	3	0	0	3	3
2	PCC	25BACCSTC08	Secure Coding	3	0	0	3	3
3	PCC	25BACCSTC09	Network Management Systems and Operations	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	25BACCSLC07	Secure Coding Laboratory	0	0	2	2	1
7	PCC	25BACCSLC08	Network Management Systems and Operations Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – III** *(Refer ANNEXURE - IV)*	1	0	2	3	2
9	INT	25BACCSIC01	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		Principles of Management	3	0	0	3	3
2	PCC	25BACCSTC10	End Point Security	2	0	0	2	2
3	PCC	25BACCSTC11	Malware Analysis	3	0	0	3	3
4	PCC	25BACCSTC12	DevOps	2	0	0	2	2
5	DE		**Discipline Elective – II** *(Refer ANNEXURE - III)*	3	0	0	3	3
6	OE		**Open Elective – III** *(Refer ANNEXURE - II)*	3	0	0	3	3
7	PCC	25BACCSLC09	End Point Security Laboratory	0	0	2	2	1
8	PCC	25BACCSLC10	DevOps 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	25BACCSTC13	Cyber Forensics	3	0	0	3	3
2	PCC	25BACCSTC14	Cloud Computing Security	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	25BACCSLC11	Cloud Computing Security Laboratory	0	0	2	2	1
8	SEC		**Skill Enhancement Course – IV** *(Refer ANNEXURE – IV)*	1	0	2	3	2
9	INT	25BACCSIC02	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	25BACCSPC01	Capstone Project	0	0	28	28	14
2	INT	25BACCSIC03	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.	25BACCSDM01	Privacy and Security in Online Social Media
2.	25BACCSDM02	Distributed Systems
3.	25BACCSDM03	Big Data Computing
4.	25BACCSDM04	Social Network Analysis
5.	25BACCSDM05	Fundamentals of Artificial Intelligence
6.	25BACCSDM06	Multi-Core Computer Architecture
7.	25BACCSDM07	Software Engineering
8.	25BACCSDM08	Compiler Construction
Any advanced Course offered by SWAYAM NPTEL can be appended in future.		

Discipline Elective – II		
S. No.	**Course Code**	**Course Title**
1.	25BACCSDC01	Operating System Security
2.	25BACCSDC02	Data Privacy and Database Security
3.	25BACCSDC03	Biometric System and Security
4.	25BACCSDC04	Applied Cryptography
Any advanced courses can be appended in future.		

Discipline Elective – III		
S. No.	**Course Code**	**Course Title**
1.	25BACCSDC05	Secure Software Engineering
2.	25BACCSDC06	Mobile Security
3.	25BACCSDC07	BlockChain Technology
4.	25BACCSDC08	Security Assessment and Risk Analysis
Any advanced courses can be appended in future.		

Discipline Elective – IV		
S. No.	**Course Code**	**Course Title**
1.	25BACCSDC09	Cyber Warfare and Defense
2.	25BACCSDC10	AI Enabled Cyber Security
3.	25BACCSDC11	Physical Security of IT Infrastructure
4.	25BACCSDC12	Social Network Security
Any advanced courses can be appended in future.		

Discipline Elective –V		
S. No.	**Course Code**	**Course Title**
1	25BACCSDC13	Access Control and Identity Management Systems
2	25BACCSDC14	Intrusion Detection and Prevention System
3	25BACCSDC15	Security Information and Event Management
4	25BACCSDC16	Cyber Laws and Security Policies
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.	25BACCSSC01	Functional Programming
Any Courses can be appended in future.		

Skill Enhancement Course – II		
S. No.	**Course Code**	**Course Title**
1.	25BACCSSC02	Full Stack Development I
Any Courses can be appended in future.		

Skill Enhancement course – III		
Sl. No.	**Course Code**	**Course Title**
1.	25BACCSSC03	Full Stack Development II
Any Courses can be appended in future.		

Skill Enhancement Course – IV		
Sl. No.	**Course Code**	**Course Title**
1.	25BACCSSC04	Penetration Testing
2.	25BACCSSC05	Ethical Hacking
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

25BACCSTC01 FUNDAMENTALS OF CYBER SECURITY

L	T	P	C
3	0	0	3

Pre-requisite: None

Course Description:
The course provides a comprehensive overview of different technical and managerial aspects related to cybersecurity. The subjects covered include risk assessment and management, cybersecurity programs, IT security controls and technologies, security standards and regulations, IT auditing, cyber insurance, and various strategies for mitigating cyber risks.

Course Objectives:
The main objectives of the course are to

1. To Understand basic concepts and the need for cyber security in the digital era.
2. To Identify different types of cyber threats, attacks, and vulnerabilities.
3. To Learn the basic principles of cryptography and authentication.
4. To Gain awareness of secure practices for devices, data, and online activities.
5. To Recognize the importance of cyber security policies and emerging trends.

UNIT I INTRODUCTION TO CYBER SECURITY **9 hours**

Introduction to Cyber Security – Need, Scope, and Applications - Types of Cyber Threats: Malware, Phishing, Ransomware, Spyware, Social Engineering - Basic Security Principles – Confidentiality, Integrity, Availability (CIA Triad) - Security Terminologies – Vulnerabilities, Risks, Attacks, Exploits - Overview of Cyber Ethics and Cyber Laws in India.

UNIT II NETWORK AND INFORMATION SECURITY BASICS **9 hours**

Basics of Computer Networks (LAN, WAN, Internet) - Network Threats: Man-in-the-Middle, DoS/DDoS, Eavesdropping, Spoofing - Firewalls – Concept and Types - Introduction to Intrusion Detection & Prevention Systems (IDS/IPS) - Basics of VPN and Proxy Servers.

UNIT III CRYPTOGRAPHY AND AUTHENTICATION **9 hours**

Introduction: Security trends, The OSI Security Architecture, Security Attacks, Security Services and Security Mechanisms, A model for Network security.

Classical Encryption Techniques: Symmetric Cipher Modes, Substitute Techniques, Transposition Techniques, Stenography.

UNIT IV SECURING DEVICES, DATA AND APPLICATIONS **9 hours**

Secure Use of Computers and Mobile Devices - Operating System Security (Windows, Linux, Android) - Safe Internet Practices – Strong Passwords, Secure Browsing, HTTPS - Email & Social Media Security - Data Backup and Recovery Methods.

UNIT V CYBER SECURITY MANAGEMENT AND TRENDS **9 hours**

Cyber Security Policies – Importance and Examples - Incident Response Basics - Risk Management Fundamentals - Introduction to Cyber Security Frameworks (ISO 27001, NIST) - Emerging Trends: Cloud Security, IoT Security.

Course Outcomes:

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

CO1: Ability to know the cyber security concepts

CO2: Understanding the different networks and threats

CO3: Integrate and apply the cryptography and authentication in network.

CO4: Distinguish knowledge of Securing Devices, Data and Applications.

CO5: Applying cyber security policies on Cyber Security frame work and Risk Management.

Text Books:

1. **"Cyber security Essentials"** – Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, Pearson, **2022**.
2. **"Introduction to Cyber security"** – Jeffrey S. Beasley, Piyasat Nilkaew, Pearson, **2nd Edition, 2023**.

Reference Books:

1. **"Principles of Computer Security: CompTIA Security+ and Beyond"** – Wm. Arthur Conklin et al., McGraw Hill, **7th Edition, 2024**.
2. **"Cyber security and Cyber war: What Everyone Needs to Know"** – P.W. Singer, Allan Friedman, Oxford University Press, **Updated Edition, 2023**.
3. **"Information Security: Principles and Practice"** – Mark Stamp, Wiley, **3rd Edition, 2022**

Mode of Evaluation: Assignments, Mid Term Tests and 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.

Text / Reference Books:

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

Mode of Evaluation: Continuous Internal Evaluation and End Semester Examination

B. Tech I Year 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: Nil

Course Description:

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

Course Objectives:

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

UNIT I OVERVIEW & INSTRUCTIONS 9 hours

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

UNIT II ARITHMETIC OPERATIONS 9 hours

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

UNIT III PROCESSOR AND CONTROL UNIT 9 hours

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

UNIT IV PARALLELISM 9 hours

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

UNIT V MEMORY AND I/O SYSTEMS 9 hours

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

Course Outcomes:

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

Text Books:

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

Reference Books:

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

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

B.Tech II Year I Semester

25BACCSTC02 DATABASE CONCEPTS AND APPLICATIONS

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Description:

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

Course Objectives:

This course enables students to

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

UNIT I **9 hours**

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

UNIT II **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 **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 **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 **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:

After completing this Unit, students will be able to

CO1: Understand the basic concepts and mechanisms 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 Book(s)

1. Silberschatz, A., Galvin, P. B., and Gagne, G., **Operating System Concepts**, 10th Edition, John Wiley & Sons, 2018.
2. Tanenbaum, A. S., and Bos, H., **Modern Operating Systems**, 5th Edition, Pearson Education, 2022.
3. Silberschatz, A., Galvin, P. B., and Gagne, G., Operating System Concepts Essentials, 3rd Edition, John Wiley & Sons, 2023.

Reference Books

1. “Operating Systems – A Spiral Approach”, Ramaz Elmasri, A. Gil Carrick, David Levine, Tata McGraw Hill Edition, 2020.
2. "Operating Systems: Internals and Design Principles", William Stallings, 7th Edition, Prentice Hall, 2023.
3. “Operating Systems”, Achyut S.Godbole, Atul Kahate, McGraw Hill Education, 2022.

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

B.Tech II Year I Semester

25BACCSTC03 MODERN OPERATING SYSTEMS

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Description:

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

Course Objectives:

This course enables students to

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

UNIT I RELATIONAL DATABASES 9 hours

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

UNIT II DATABASE DESIGN 9 hours

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

UNIT III TRANSACTION PROCESSING 9 hours

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

UNIT IV IMPLEMENTATION TECHNIQUES 9 hours

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

UNIT V ADVANCED APPLICATION DEVELOPMENT AND RECENT TREND 9 hours

Advanced Application Development – Connecting to MongoDB with Python, MongoDB query Language, Updating/Deleting documents in collection, MongoDB query operators. Distributed Databases: Architecture– NOSQL Databases: Introduction – CAP Theorem – Document Based systems – Key value Stores – Column Based Systems – Graph Databases. Database Security:

Security issues – Access control based on privileges – Role Based access control – SQL Injection – Statistical Database security – Flow control – Encryption and Public Key infrastructures – Challenges

Course Outcomes:

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

Text Book(s)

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

Reference Books

1. C.J.Date, A.Kannan, S.Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2022.
2. Raghu Ramakrishnan and Johannas Gehrke, "Database Management Systems", Eighth Edition, McGraw-Hill, 2023.

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

B. Tech II Year I Semester

25BACCSTC04 DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS

L	T	P	C
3	0	0	3

Pre-requisite: Nil

Course Description:

This course introduces fundamental concepts of algorithm design and analysis, focusing on efficiency and complexity evaluation. It covers hierarchical data structures, graph algorithms, and algorithm design strategies such as dynamic programming and greedy methods. The course also provides insights into NP-complete problems, enabling students to select and design appropriate algorithms for real-world computational challenges

Course Objectives:

This course enables students to

1. Understand and analyze algorithms and compute time and space complexity for efficient problem solving.
2. Apply hierarchical data structures such as trees, heaps, and disjoint sets in computing applications.
3. Analyze and implement graph-based algorithms to solve real-world problems.
4. Design and select appropriate data structures and algorithmic strategies for given problems.
5. Evaluate computational problems to identify NP-complete characteristics and apply suitable solution approaches.

UNIT I COMPLEXITY ANALYSIS 9 hours

Algorithms – Algorithms as a Technology -Time and Space complexity of algorithms-Asymptotic analysis-Average and worst-case analysis-Asymptotic Notation-Importance of efficient algorithms Program performance measurement

Recurrences: The Substitution Method – The Recursion Tree Method- Masters Method –

Examples algorithm analysis: Searching algorithms – Linear Search – Binary Search. Sorting Algorithms: Bubble sort, Selection sort, Insertion sort.

UNIT II DIVIDE-AND-CONQUER AND GREEDY METHOD 9 hours

Divide-and-Conquer: Introduction - Binary search, Sorting - Quick sort, Merge sort, Min-Max problem - Strassen's matrix multiplication.

Greedy Method: Introduction – Minimum Spanning Trees, Prim's and Kruskal's Algorithms – The Knapsack Problem – Job Scheduling problem – Single Source Shortest paths, Dijkstra's Algorithm – Huffman trees and Coding.

UNIT III DYNAMIC PROGRAMMING AND BRUTE FORCE 9 hours

Dynamic Programming: Introduction - Optimal binary search trees - 0/1 Knapsack problem - All pairs shortest path problem - Matrix chain multiplication problem - Travelling salesman problem, Reliability design.

Brute Force: Closest-Pair and Convex-Hull Problems - Exhaustive Search - Traveling Salesman Problem - Knapsack Problem - Assignment problem.

UNIT IV BACKTRACKING AND BRANCH-AND-BOUND 9 hours

Backtracking: Introduction – Applications - N-queen's problem, Sum of subsets problem, Graph coloring and Hamiltonian cycles.

Branch-and-Bound: Introduction - LC Branch and Bound solution - FIFO Branch and Bound solution - 0/1 Knapsack Problem - Travelling Salesman problem.

UNIT V GRAPH ALGORITHMS AND NP COMPLETENESS 9 hours

Graph Algorithms: Representations of Graphs – Breadth-First Search – Depth-First Search – Topological Sort – Strongly Connected Components- Single-Source Shortest Paths: The Bellman-Ford algorithm.

NP-Completeness: Polynomial Time – Polynomial-Time Verification – NP- Completeness and Reducibility – NP-Completeness Proofs – NP-Complete Problems, Application of Boolean Satisfiability (SAT) and Graph Coloring in NP-Complete Problem Solving.

Course Outcomes:

After completing this course, students will be able to

CO1: Design algorithms and find suitable data structures to solve computing problems with and without performance constraint

CO2: Reformulate problem statements and apply divide-and-conquer and greedy strategies to obtain efficient solutions.

CO3: Evaluate and apply suitable algorithm design strategies such as dynamic programming and brute force methods for effective problem solving.

CO4: Design and develop algorithms independently for backtracking and branch-and-bound problems.

CO5: Apply graph-based algorithms to solve real-world computational problems.

Text Book(s)

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., Introduction to Algorithms, MIT Press, 4th Edition, 2022.
2. Levitin, A., Introduction to the Design and Analysis of Algorithms, Pearson Education, 4th Edition, 2024.
3. Kleinberg, J., and Tardos, É., Algorithm Design, 1st Edition, Pearson Education, 2005 (Pearson eTextbook Update, 2021).

Reference Books

1. Weiss, M. A., *Data Structures and Algorithm Analysis in C++*, Pearson Education, 4th Edition, 2014.
2. Skiena, S. S., *The Algorithm Design Manual*, Springer Nature, 3rd Edition, 2020.
3. Sedgewick, R., and Wayne, K., *Algorithms*, Addison-Wesley Professional, 4th Edition, 2011 (Updated Pearson eText Release 2021–22).

4. Roughgarden, T., *Algorithms Illuminated Series (Volumes 1–4)*, Soundlikeyourself Publishing, 2020.
5. Erickson, J., *Algorithms*, Open Access Edition, Updated 2024.

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

B. Tech II Year I Semester

25BACCSLC01 DATABASE CONCEPTS AND APPLICATION 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:

This course enables students to

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 Experiments:

1. **Entity Relationship (ER) Modeling and Database Design**
   - Design ER diagrams for Library Management System and Hospital Management System.
2. **Database Creation and Data Definition Language (DDL) Operations**
   - Creation, alteration, renaming, copying, and deletion of database tables.
3. **Data Manipulation Language (DML) Operations**
   - Insert, update, delete, and retrieve records from relational databases.
4. **Basic SQL Query Processing**
   - Retrieval of data using selection, projection, aliases, and filtering conditions.
5. **Pattern Matching and String Search using LIKE Predicate**
   - Implementation of wildcard searches and pattern matching techniques.
6. **Aggregate Functions and Data Sorting**
   - Application of SUM, AVG, COUNT, MAX, MIN, ORDER BY, and GROUP BY clauses.
7. **Integrity Constraints and Database Validation**
   - Implementation and verification of Primary Key, Foreign Key, Unique, Check, and Referential Integrity Constraints.
8. **Single Row Functions and Date Functions**
   - Usage of character, numeric, conversion, and date functions in SQL.
9. **Join Operations on Multiple Tables**
   - Implementation of Inner Join, Outer Join, Self Join, and Equi Join queries.
10. **Group Functions and Aggregation Analysis**
    - Advanced grouping, HAVING clause, and departmental summary analysis.

11. **Nested Queries and Subquery Processing**
    - Single-row, multi-row, correlated, and nested subqueries.
12. **PL/SQL Programming Fundamentals**
    - Development of PL/SQL blocks, variables, conditional statements, and loops.
13. **Cursor Management in PL/SQL**
    - Implementation of explicit and implicit cursors for record processing.
14. **Stored Procedures and Database Triggers**
    - Creation of procedures, functions, and triggers for database automation.
15. **Database Connectivity using JDBC**
    - Java-Oracle database connectivity for CRUD operations using JDBC and Prepared Statements.
16. **NoSQL Database Design and Application Development**
    - Design and implementation of NoSQL-based Student Registration and Employee Management Systems.
17. **Mini Project: Student Information Management System**
    - ER Modeling
    - Database Design and Normalization
    - Table Creation and Relationship Establishment
    - Front-End GUI Development
    - Oracle Database Connectivity
    - Project Documentation and Demonstration

Course Outcomes:

After 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.

References 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 Semester End Examination.

B. Tech II Year I Semester

25BACCSLC02 MODERN OPERATING SYSTEMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite Nil

Course Description:

This laboratory course provides hands-on experience in understanding and implementing fundamental operating system concepts. Students explore key topics such as process management, CPU scheduling, deadlock handling, memory management, and file system operations through simulations and programming exercises. The course also introduces the Unix/Linux environment, including essential commands and system utilities. Emphasis is placed on system-level programming using C and Unix system calls for process creation, inter-process communication, and input/output management. Through practical experiments, students develop the skills required to analyze, implement, and evaluate operating system mechanisms used in modern computing systems.

Course Objectives:

This course enables students to

1. To develop practical knowledge of fundamental operating system concepts through simulation and implementation of core OS functionalities.
2. To understand process management techniques, including process creation, scheduling, synchronization, and inter-process communication.
3. To gain hands-on experience with memory management, deadlock handling, and file system operations in operating systems.
4. To familiarize students with the Unix/Linux environment, including command-line utilities and system administration basics.
5. To enable the development of C programs utilizing Unix system calls for process control, communication, and input/output operations.

List of Experiments:

1. Study and execute basic UNIX/Linux commands for file handling, directory management, and process monitoring.
2. Write a C program using UNIX/Linux system calls such as open(), read(), write(), close(), and lseek() for file operations.
3. Write a C program using process-related system calls such as fork(), exec(), wait(), and exit().
4. Implement the First Come First Serve (FCFS) CPU Scheduling algorithm and analyze its performance.
5. Implement the Shortest Job First (SJF) CPU Scheduling algorithm and compare it with FCFS.
6. Implement the Priority CPU Scheduling algorithm and evaluate scheduling efficiency.
7. Implement the Round Robin CPU Scheduling algorithm for different time quantum values.
8. Simulate the Banker's Algorithm for deadlock avoidance and safe-state determination.
9. Implement the Producer–Consumer problem using semaphores and UNIX/Linux system calls.
10. Develop programs to demonstrate Inter-Process Communication (IPC) using Pipes and FIFOs.
11. Develop programs to demonstrate Inter-Process Communication (IPC) using Message Queues and

Shared Memory.

12. Simulate contiguous memory allocation techniques such as First Fit, Best Fit, and Worst Fit.
13. Implement memory management techniques using Paging and Segmentation schemes.
14. Simulate page replacement algorithms such as FIFO and LRU, and compare their page fault rates.
15. Simulate the Optimal Page Replacement algorithm and compare its performance with FIFO and LRU algorithms.

Course Outcomes:

After completion of the course, students will be able to

CO1: Explain the fundamental concepts of operating systems, including process management, scheduling, synchronization, and resource allocation mechanisms.

CO2: Simulate and implement operating system algorithms related to CPU scheduling, deadlock management, memory management, and file systems.

CO3: Analyze the behavior and performance of operating system components using appropriate simulation and experimental techniques.

CO4: Apply Unix/Linux commands and system calls to develop programs for process creation, inter-process communication, and I/O management.

CO5: Evaluate and design efficient solutions to operating system problems through practical implementation and performance assessment.

Text Books:

1. Silberschatz, A., Galvin, P. B., and Gagne, G., **Operating System Concepts**, 10th Edition, John Wiley & Sons, 2018.
2. Stevens, W. R., and Rago, S. A., **Advanced Programming in the UNIX Environment**, 3rd Edition, Addison-Wesley Professional, 2013.

References Books:

1. Stallings, W., **Operating Systems: Internals and Design Principles**, 10th Edition, Pearson Education, 2024.
2. Tanenbaum, A. S., and Bos, H., **Modern Operating Systems**, 5th Edition, Pearson Education, 2022.
3. Anderson, T., and Dahlin, M., **Operating Systems: Principles and Practice**, 2nd Edition, Recursive Books, 2020.
4. Vahalia, U., **UNIX Internals: The New Frontiers**, Pearson Education, Reprint Edition, 2017.
5. Kerrisk, M., **The Linux Programming Interface**, No Starch Press, Reprint Edition, 2024.

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

B. Tech II Year I Semester

25BACCSLC03 DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite:

Basic knowledge of programming, data structures, mathematical foundations, recursion, problem-solving techniques, and proficiency in C/C++ programming language.

Course Description:

The Design and Analysis of Algorithms Laboratory provides hands-on experience in implementing and evaluating fundamental algorithms. Students develop solutions using sorting, searching, divide-and-conquer, greedy, dynamic programming, backtracking, and graph algorithms. The course emphasizes algorithm design, complexity analysis, performance evaluation, and problem-solving skills, enabling students to develop efficient and optimized solutions for real-world computational challenges.

Course Objectives:

This course enables students to

1. To develop proficiency in implementing fundamental algorithms and data processing techniques using programming languages.
2. To understand and apply various algorithm design paradigms such as divide-and-conquer, greedy, dynamic programming, and backtracking.
3. To analyze and compare algorithms based on their time and space complexity for efficient problem solving.
4. To design and implement graph, searching, sorting, and optimization algorithms for computational problems.
5. To enhance problem-solving, analytical thinking, and programming skills through practical experimentation and performance evaluation of algorithms.

List of Experiments:

1. Analyze the Performance of Linear Search and Binary Search Algorithms
2. Evaluate Sorting Techniques: Bubble, Selection, and Insertion Sort
3. Analyze Merge Sort using the Divide-and-Conquer Strategy
4. Evaluate the Performance of Quick Sort and Merge Sort
5. Analyze Heap Sort and its Computational Efficiency
6. Apply Recursive Techniques to Solve Classical Problems
7. Develop and Compare Strassen's Matrix Multiplication Algorithm
8. Evaluate the Maximum–Minimum Problem using Divide-and-Conquer
9. Analyze the Fractional Knapsack Problem using the Greedy Method
10. Evaluate Job Sequencing with Deadlines using a Greedy Approach
11. Compare Minimum Spanning Tree Algorithms: Prim's and Kruskal's
12. Analyze Dijkstra's Single Source Shortest Path Algorithm
13. Design and Evaluate Huffman Coding for Data Compression

14. Analyze Matrix Chain Multiplication using Dynamic Programming
15. Evaluate the 0/1 Knapsack Problem using Dynamic Programming
16. Analyze the Longest Common Subsequence Algorithm
17. Develop a Solution for the N-Queens Problem using Backtracking
18. Design and Analyze the Graph Coloring Problem using Backtracking
19. Develop and Evaluate Solutions for the Travelling Salesman Problem
20. Compare and Evaluate Branch-and-Bound Techniques for Optimization Problems

Course Outcomes:

After completion of the course, students will be able to

CO1: Explain fundamental algorithm design techniques, complexity measures, and performance evaluation methods.

CO2: Analyze computational problems and select appropriate algorithmic strategies such as divide-and-conquer, greedy, and dynamic programming.

CO3: Implement searching, sorting, graph, and optimization algorithms using suitable programming languages.

CO4: Evaluate and compare algorithms based on their time and space complexities to determine efficient solutions.

CO5: Design and develop optimized algorithmic solutions for real-world computational problems using advanced design paradigms.

Text Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, 4th Edition, MIT Press, 2022.
2. Algorithm Design, Jon Kleinberg and Éva Tardos, 1st Edition, Pearson Education, 2006. (Reprint/Updated eText releases available through 2021–2022).
3. Fundamentals of Computer Algorithms, Universities Press, 2nd Edition, 2008.

References Books:

1. The Algorithm Design Manual, Steven S. Skiena, 3rd Edition, Springer, 2020.
2. Algorithms, Robert Sedgewick and Kevin Wayne, 4th Edition, Addison-Wesley, 2011.
3. Design and Analysis of Algorithms, S. Sridhar, Oxford University Press, 2014.
4. Computer Algorithms, Sara Baase and Allen Van Gelder, 3rd Edition, Pearson Education, 2000.
5. Data Structures and Algorithm Analysis in C++, Mark Allen Weiss, 4th Edition, Pearson, 2014.
6. Algorithms in C, Robert Sedgewick, Addison-Wesley Professional, 2002.

Mode of Evaluation: Continuous Internal Evaluation and Semester End 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

25BACCSTC05 PRINCIPLES OF OBJECT ORIENTED PROGRAMMING

L	T	P	C
3	0	0	3

Pre-requisite Course on programming for problem solving.

Course Description:

This course introduces the fundamentals of object-oriented programming by explaining the differences between procedural and object-oriented approaches and core OOP concepts such as objects, encapsulation, and abstraction. It focuses on modeling real-world problems using classes, inheritance, polymorphism, and UML diagrams, while applying these concepts through Java programming. The course also emphasizes developing reliable and reusable Java applications using inheritance, interfaces, exception handling, collections, and basic memory management.

Course Objectives:

1. To understand the difference between procedural programming and object-oriented programming, and learn basic OOP concepts like objects, encapsulation, and abstraction.
2. To identify and model real-world problems using OOP concepts such as classes, objects, inheritance, polymorphism, and UML diagrams.
3. To learn and apply basic Java programming concepts, including creating classes, using constructors, access modifiers, packages, and encapsulation.
4. To develop Java programs using inheritance and polymorphism, along with interfaces, abstract classes, and method overriding.
5. To write reliable Java programs by handling errors and managing resources, using exception handling, collections, strings, and memory management concepts.

UNIT I FOUNDATIONS OF OBJECT-ORIENTED DESIGN 9 hours

Paradigm Shift: Limitations of Structured Programming; Data-centric vs. Procedure-centric design.
Core Concepts: Definition of Objects (State + Behavior) and Classes (Blueprints).
Encapsulation: The "Black Box" principle; Information Hiding and Interface vs. Implementation.
Abstraction: Managing complexity through essential features; Generalization vs. Specialization.

UNIT II ADVANCED OOP PRINCIPLES 9 hours

Inheritance: "Is-A" relationships; Base classes, Derived classes, and Hierarchical classification.
Polymorphism: Concept of "One Interface, Many Forms"; Static vs. Dynamic binding. Object
Relationships: * Aggregation & Composition: "Has-A" and "Part-Of" relationships.
Dependency: Using one object within another.

UNIT III INTRODUCTION TO JAVA & ENCAPSULATION 9 hours

Java Ecosystem: JVM Architecture, Bytecode, and Platform Independence.
Language Basics: Primitive types vs. Reference types, Arrays, and Control Flow.
Implementing Classes: Java syntax for classes, methods, and this keyword.
Constructors: Memory allocation and initialization; Overloading. Access Control: public, private, protected, and default modifiers.
Packages: Organizing code and managing namespaces.

UNIT IV CLASSES AND OBJECTS IN JAVA 9 hours

Extending Classes: The extends keyword and the super keyword**.**
Method Overriding: Runtime polymorphism and Dynamic Method Dispatch.
Abstract Classes: Implementing partial abstraction in Java.
Interfaces: Complete abstraction; implementing multiple interfaces and overcoming the "Diamond Problem."
Final Keyword: Creating constants and preventing inheritance/overriding.

UNIT V Robustness and Resource Management 9 hours

Exception Handling: The try-catch-finally block; Checked vs. Unchecked exceptions.
Custom Exceptions: Defining and throwing domain-specific errors.
String Management: Memory efficiency with String, StringBuilder, and the String Pool.
Collection Framework: Introduction to ArrayList, HashSet, and HashMap.
Memory Management: Basics of Garbage Collection and the Object class.

Course Outcomes:

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

CO1: Explain the difference between procedural and object-oriented paradigms.

CO2: Deconstruct complex real-world systems into interacting objects and classes.

CO3: Create UML Class Diagrams to represent system architecture.

CO4 : Develop functional Java applications that utilize inheritance and polymorphism.

CO5: Implement error-handling strategies to ensure software reliability.

Text Book(s)

1. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, 2nd Edition, Pearson Education, 2017
2. Programming With Java by E. Balagurusamy, McGraw-Hill Education — 7th Edition (2023).

Reference Books

1. Robert C. Martin, Clean Architecture: A Craftsman’s Guide to Software Structure and Design Pearson Education, 2018
2. Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, Pearson Education, 2017
3. Herbert Schildt & Danny Coward, Java: The Complete Reference,13th Edition, McGraw-Hill Education, 2024
4. Cay S. Horstmann, *Programming with Java: Beginner to Advanced* (Indian adaptation) 7th Edition, Wiley India, 2024

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

B. Tech II Year II Semester

25BACCSTC06 COMPUTER NETWORKS

L	T	P	C
3	0	0	3

Pre-requisite Nil

Course Description:

This course provides a comprehensive introduction to computer networks and data communications, focusing on the principles, architectures, protocols, and services that enable reliable data exchange across interconnected systems. By the end of the course, students will gain a solid foundation in network design, protocol operation, and real-world networking applications, enabling them to understand and analyze modern communication networks.

Course Objectives:

This course enables students to

1. Understand the fundamentals of data communication, networking concepts, and protocol layering.
2. Know the functions of protocols of each layer of TCP/IP protocol suite.
3. Visualize the end-to-end flow of information.
4. Learn the functions of network layer, transport layer and the various routing protocols.
5. Learn the working of application layer protocols and evaluate real-world network in applications.

UNIT I NETWORK PRELIMINARIES AND PHYSICAL LAYER 9 hours

Data Communication : Networks & Topologies - Network Types, **Layering principles:** TCP/IP Reference Model- Protocol Suite, **Physical Layer**: Data and Signals- Encoding Schemes -Unipolar NRZ- Polar NRZ-L- Polar NRZ-I- Return to Zero - Manchester – Performance, **Transmission Media: Guided Media**: CAT5,CAT6 Twisted-Pair Cable- Optical Fiber Cables , E**thernet Standard:** 10G,!00G,400G, **Unguided Media**: Radio Waves-Microwaves- Infrared-Satellite Communication

UNIT II DATA LINK LAYER 10 hours

Data Link Layer Introduction, Framing : Character count, byte stuffing, bit stuffing, physical layer coding violations, **Error Control :**Error detection and correction techniques, parity check, checksum, cyclic redundancy check (CRC), retransmission mechanisms, **Data Link Layer Protocols :** Simple protocols, Stop-and-Wait ARQ, Sliding Window ARQ, **Sliding Window ARQ :** Go-Back-N, Selective Repeat, **HDLC :** Frame structure, modes of operation, error and flow control, **PPP :** Frame format, phases of operation, authentication (PAP, CHAP), **MAC Sublayer :** Channel allocation, multiple access principles, **Random Access Protocols :** ALOHA, Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA, **Controlled Access Protocols :** Reservation, polling, token passing, **Ethernet :** IEEE 802.3, frame format, MAC addressing, Ethernet operation, **Virtual LAN (VLAN) :** VLAN concepts, IEEE 802.1Q, VLAN tagging, advantages, **Wireless LAN :** IEEE 802.11 architecture, MAC protocol, frame types, CSMA/CA,

UNIT III NETWORK LAYER 10 hours

Network Layer Introduction, Design Issues : Store-and-forward packet switching, services provided to transport layer, connectionless and connection-oriented services, **Logical Addressing :** IPv4 addressing, classes of IP addresses, special addresses, Subnetting, Supernetting, Classless Inter-Domain Routing (CIDR), **Address Resolution :** ARP, RARP, ICMP, **Routing Algorithms** : Shortest path routing, Distance Vector routing (Bellman-Ford), Link State routing, Flooding, Hierarchical routing, **Congestion Control Algorithms** :Traffic shaping algorithms (Leaky Bucket, Token Bucket, Choke Packet) admission control, **Quality of Service (QoS)** : Bandwidth, delay, jitter, packet loss, scheduling and buffering, **Internetworking** : Routers, gateways, packet fragmentation and reassembly, Tunneling,

UNIT IV TRANSPORT LAYER 10 hours

Transport service - Elements of transport protocols ,User Datagram Protocol, **Transmission Control Protocol:** 3 way handshaking, Connection Establishment & Release, Timers in TCP, Timeout Complications in TCP, Multiplexing in TCP, Applications of TCP in evaluating a design.

UNIT V APPLICATION LAYER 6 hours

Application Layer Introduction, Network applications and application layer protocols, Client–Server and Peer-to-Peer architectures, Socket programming concepts, Transport services for applications, **Domain Name System (DNS) :** Name space, resource records, message format, name resolution, **Electronic Mail :** Architecture, SMTP, POP3, IMAP, **World Wide Web (WWW) :** HTTP, persistent and non-persistent connections, cookies, caching, Web proxies, **File Transfer Protocol (FTP) :** File transfer mechanisms, control and data connections, **Remote Login :** TELNET, Secure Shell (SSH), **Network Management :** SNMP basics, **Multimedia Applications :** Streaming, real-time protocols (RTP, RTCP), Content Distribution Networks (CDN), **Security at Application Layer :** HTTPS.

Course Outcomes:

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

CO1: Learn basics of networks, topologies, TCP/IP and OSI models, signals, encoding schemes, transmission media, and Ethernet standards.

CO2: Understand framing, error detection and correction, ARQ protocols, MAC sublayer functions, Ethernet, VLANs, and wireless LAN operations.

CO3: Design networks, assign addressing schemes, and develop routing tables to ensure efficient communication.

CO4: Analyze application requirements and select suitable transport layer protocols (TCP/UDP) for implementation.

CO5: Understand and use application layer protocols and services for web, email, file transfer, and multimedia.

Text Book(s)

1. Tanenbaum, A. S., Feamster, N., and Wetherall, D. J., Computer Networks, 6th Edition, Pearson Education, 2021.
2. Behrouz A. Forouzan, Data Communications and Networking with TCP/IP Protocol Suite, Sixth Edition TMH, 2022.
3. James F. Kurose, Keith W. Ross, Computer Networking, A Top-Down Approach Featuring the Internet, Eighth Edition, Pearson Education, 2021.

4. Kurose, J. F., and Ross, K. W., Computer Networking: A Top-Down Approach, 8th Edition, Pearson, 2021.

Reference Books

1. Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, Fifth Edition, Morgan Kaufmann Publishers Inc., 2012.
2. William Stallings, Data and Computer Communications, Tenth Edition, Pearson Education, 2013.

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

B. Tech II Year II Semester

25BACCSTC07 CRYPTOGRAPHY

	L	T	P	C
Prerequisite: 25BAMATTC08	**3**	**0**	**0**	**3**

Course Description:
A comprehensive study of classical and modern cryptographic techniques including block ciphers, public-key cryptography, hash functions, and digital signatures. The course also explores practical security applications such as Kerberos, IPSec, SSL/TLS, Blockchain, and IoT security.

Course Objectives:
Students will be able to:

1. To comprehend the essential principles of cryptography, cryptanalysis, and cryptology, along with their significance in safeguarding communication systems.
2. To understand and assess the fundamentals of block ciphers and their operational modes.
3. To grasp and implement public key cryptographic methods.
4. To understand and appraise authentication systems utilizing hash functions.
5. To employ authentication and network security protocols.

UNIT I FOUNDATIONS OF CRYPTOGRAPHY 9 hours

Basic Concepts: Cryptography, cryptanalysis, cryptology.
Classical Cryptosystems: shift cipher, affine cipher, Vignere cipher, substitution, transposition techniques.
Fundamental Cryptographic Operations: XOR / Modulo-2 (Binary) Addition.
Modern Cryptography Foundations: Feistel Structure, Block cipher principles, Block cipher modes of operation.

UNIT II BLOCK CIPHERS AND MODES OF OPERATIONS 9 hours

DES - Data Encryption Standard-Block AES, Double and Triple DES-Blowfish-RC5.

UNIT III PUBLIC KEY CRYPTOGRAPHY 9 hours

Public Key Cryptosystem, Key distribution, Diffie Hellman Key Exchange-MITM Attack - RSA, Random Number Generation. Key Management, Password Protocol- PAP, CHAP, MS-CHAP.

UNIT IV HASH FUNCTIONS AND DIGITAL SIGNATURES 9 hours

Authentication requirement– Authentication function – MAC – Hash function – SHA - HMAC - Digital Signature Standards (DSS) and authentication protocols.

UNIT V APPLICATIONS 9 hours

Authentication – Password-Based Authentication, Multi-Factor Authentication (MFA), IP Security – IPSec, Email Security-S/MIME, Email Authentication Mechanisms- SPF, DKIM, DMARC**,** Web Security - SSL, TLS, IoT Security.

Course Outcomes:

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

CO1: Understand the foundational concepts of cryptography, cryptanalysis, and classical encryption techniques such as Shift, Affine, Vigenère, substitution, and transposition ciphers.
CO2: Analyze and apply symmetric-key block cipher algorithms including DES, AES, 3DES, Blowfish, and RC5 along with various block cipher modes of operation.
CO3: Apply public-key cryptographic techniques such as Diffie–Hellman, RSA, ECC, secure

random number generation, and key management, and evaluate related security challenges.

CO4: Evaluate the role of hash functions, MAC, HMAC, SHA algorithms, and digital signatures in ensuring message integrity and authentication.

CO5: Implement and assess security mechanisms and protocols used in real-world applications, including Kerberos, IPSec, SSL/TLS, Blockchain, and IoT security.

Text Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice –8th Edition, Pearson Education,2023.
2. Behrouz A. Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, 3rd Edition, Mc Graw Hill Education.

Reference Books:

1. Rich Helton, Johennie Helton, Mastering Java Security: Cryptography Algorithms and Practices, John Wiley Publishers.
2. Charles P. Pleeger, “Security in Computing”, Pearson Education Asia, 5th Edition.
3. William Stallings, “Network Security Essentials: Applications and standards”, Person Education Asia.
4. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security: Private Communication in a public world”, Prentice Hall India, 6nd Edition. Hill, 2013

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

B. Tech II Year II Semester

25BACCSLC04 PRINCIPLES OF OBJECT ORIENTED PROGRAMMING LABORATORY

L	T	P	C
0	0	2	1

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

Course Description:

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

Course Objectives:

This course enables students to

1. Understand and apply object-oriented analysis and design principles using real-world problem modeling and UML diagrams.
2. Translate UML designs into Java programs, demonstrating correct use of classes, objects, and relationships.
3. Implement core object-oriented concepts such as encapsulation, inheritance, polymorphism, abstraction, and interfaces in Java.
4. Develop reliable and maintainable Java applications using exception handling, collections, and multithreading.
5. Design and implement a complete mini-project following industry-oriented OOAD and Java programming practices.

List of Experiments:

Phase 1: Object-Oriented Analysis & Design (Units 1-2)

In this phase, students use pen-and-paper or modeling tools (like StarUML or LucidChart) to master the "Mindset."

1: Real-World Object Modeling

- **Task:** Identify entities, attributes, and behaviors for a "University Management System."
- **Exercise:** Differentiate between a Class (Blueprints) and an Object (Instances).

2: UML Class Diagrams (Relationships)

- **Task:** Draw a UML Class Diagram for an "E-commerce System."
- **Exercise:** Visualize **Inheritance** (User → Customer/Admin), **Aggregation** (Cart contains Products), and **Composition** (Order contains OrderItems).

3: Generalization and Specialization

- **Task:** Refine a "Payment System" model.
- **Exercise:** Use Abstraction to define a generic Payment class and specialized CreditCard, UPI, and Cash classes.

Phase 2: Java Implementation (Units 3-4)

Transitioning from diagrams to executable code.

4: Setting up the Environment & Basic Syntax

- **Task:** Install JDK and configure IDE (IntelliJ/Eclipse/VS Code).
- **Exercise:** Write a program to find the prime numbers in a range using Java control structures.

5: Encapsulation & Constructors
- **Task:** Implement a BankAccount class.
- **Exercise:** Use private variables, public getters/setters, and multiple constructors (Default vs. Parameterized).

6: Static vs. Instance Members
- **Task:** Create a Student class where collegeName is static and studentName is instance-based.
- **Exercise:** Count the total number of objects created using a static counter.

7: Inheritance & Super Keyword
- **Task:** Implement the "Employee" hierarchy designed in Week 2.
- **Exercise:** Use the super keyword to invoke parent constructors and methods.

8: Polymorphism (Static vs. Dynamic)
- **Task:** Method Overloading (Calculate area of different shapes) and Method Overriding (Animal sounds).
- **Exercise:** Demonstrate "Late Binding" using a parent reference pointing to a child object.

9: Abstraction & Interfaces
- **Task:** Convert the "Payment System" UML (Week 3) into code.
- **Exercise:** Use an Interface to define standard behaviors and implement them across different classes.

10: String Handling & Memory
- **Task:** Compare String and StringBuilder performance.
- **Exercise:** Write a program to reverse words in a sentence and check for anagrams.

11: Exception Handling (Standard)
- **Task:** Create a "Calculator" that handles Arithmetic Exception and Input Mismatch Exception.
- **Exercise: Use the finally block to ensure resource closure.**

12: Custom Exceptions
- **Task:** Create an "Age Validation" system for a voting app.
- **Exercise:** Throw a user-defined InvalidAgeException.

13: Data Structures (Collections)
- **Task:** Implement a "Phonebook" application.
- **Exercise:** Use ArrayList to store contacts and HashMap to search by name.

14: Multithreading
- **Task:** Create a "Download Manager" simulation.
- **Exercise:** Use multiple threads to simulate downloading files simultaneously.

15: Mini-Project Presentation
- **Task:** Implement a small system (Library, Pharmacy, or Hotel Management) from scratch.
- **Requirements:** Must include a UML diagram, Inheritance, and Exception Handling.

Software tools:
StarUML / Lucidchart / Draw.io
JDK 8 or above (Recommended: JDK 11 / JDK 17)

Course Outcomes:

After completion of the course, students will be able to

CO1: Analyze real-world problems and model them using object-oriented concepts such as classes, objects, attributes, and behaviors with UML diagrams.

CO2: Design object-oriented systems by applying inheritance, aggregation, composition, and abstraction principles.

CO3: Implement object-oriented applications in Java using encapsulation, constructors, static members, and polymorphism.

CO4: Develop robust Java programs using exception handling, strings, collections, and multithreading.

CO5: Design and implement a mini-project by integrating UML design, Java programming, inheritance, and exception handling principles.

Reference Books:

1. Object-Oriented Analysis and Design with Applications **-** Grady Booch, 3rd Edition, Addison-Wesley, **2022**.
2. **Core Java Volume I – Fundamentals -** Cay S. Horstmann, **12th Edition**, Pearson Education, **2024**.

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

B. Tech II Year II Semester

25BACCSLC05 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 and represent network structures and implement graph-based algorithms.
2. To design and develop TCP and UDP client-server applications for reliable communication.
3. To enable multiple client communication and handle concurrent connections efficiently.
4. To study and apply error detection techniques and routing protocols in networks.
5. To configure wired and wireless networks, including VLANs, routers, and web-based communication.

List of Experiments:

Representation of Graphs and Shortest Path Algorithms **(2 Weeks)**
Implementation of graph representations using Adjacency Matrix and Adjacency List.
Implementation of Shortest Path Algorithms: Bellman–Ford Algorithm and Dijkstra's Algorithm.
Performance analysis and comparison of Bellman–Ford and Dijkstra's algorithms.

Implementation of Client–Server Socket Programming Using TCP (1 Week)
Creation of TCP sockets
Binding a socket to an IP address and port number
Listening for incoming client connection requests
Establishing a connection between client and server
Understanding blocking and non-blocking socket modes
Sending and receiving data between client and server (Echo Server)
Implementation of reliable client–server communication using TCP

Implementation of Client–Server Socket Programming Using UDP (1 Week)
Creation of UDP sockets
Binding the socket to an IP address and port number
Understanding connectionless communication
Sending and receiving datagrams using UDP
Implementation of a UDP Echo Server
Understanding blocking and non-blocking modes in UDP sockets
Implementation of client–server communication using UDP

Multiple Client Communication and Multiplexed I/O Operations (2 Weeks)
Design and implement multiple client communication using TCP and UDP sockets.
Study and Evaluation of multiplexed I/O operations using select() and poll() for efficient handling of multiple sockets.

Study of CRC-based error detection for reliable data communication. (2 Weeks)
Study of polynomial representation in error detection and correction mechanisms.
Detailed analysis of Cyclic Redundancy Check (CRC) technique.
Representation of data and generator polynomials in CRC polynomial form.
Generation of CRC codewords at the sender side.
Detection of transmission errors at the receiver using CRC verification.
Evaluation of the effectiveness of CRC in ensuring reliable data transmission.

Simulation and performance analysis of Link State and Distance Vector routing protocols using NS2. (1 Week)
Design and creation of network scenarios using the NS2 simulation environment
Simulation of Link State Routing Protocol.
Simulation of Distance Vector Routing Protocol
Comparison of routing paths selected by Link State and Distance Vector routing algorithms.

Creation of routing tables and configuration of routers for wired and wireless networks, including **VLAN setup and SSID configuration**. **(2 Weeks)**
Design and implement routing tables for network communication.
Configure routers for: Wired networks & Wireless networks
Design and implement VLAN creation and configuration.
Verify routing tables to ensure correct paths.
Configure and simulate SSID settings for wireless networks.

Implement and understand TCP and UDP client-server communication using web-based interfaces. **(1 Week)**
Implement TCP and UDP client–server communication using web interfaces.
Implementation of a TCP Echo Client–Server and UDP Echo Client–Server using java.net packages.
Enable multiple clients to interact with servers in real time through a browser.
Visualize and analyze server responses for connection-oriented (TCP) and connectionless (UDP) protocols.

Course Outcomes:
After completion of the course, students will be able to
CO1: Represent and analyze networks and implement shortest path algorithms.
CO2: Design and implement TCP and UDP client-server communication.
CO3: Enable multiple client communication and handle concurrent connections.
CO4: Apply error detection techniques and simulate routing protocols.
CO5: Configure networks, routers, VLANs, and web-based multi-client communication.

Text Books:

1. Stevens, W. R., Fenner, B., and Rudoff, A. M., *UNIX Network Programming, Volume 1: The Sockets Networking API*, 3rd Edition, Addison-Wesley Professional, 2004.
2. Stevens, W. R., and Rago, S. A., *UNIX Network Programming, Volume 2: Interprocess Communications*, 2nd Edition, Addison-Wesley Professional, 2004.
3. Kerrisk, M., *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*, No Starch Press, 2024 Reprint Edition.
4. Donahoo, M. J., and Calvert, K. L., *TCP/IP Sockets in C: Practical Guide for Programmers*, 3rd Edition, Morgan Kaufmann, 2021.

5. Stevens, W. R., and Fall, K. R., *TCP/IP Illustrated, Volume 1: The Protocols*, 2nd Edition, Addison-Wesley Professional, 2011.

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, Model Test and End Semester Examination

B. Tech II Year II Semester

25BACCSLC06 CRYPTOGRAPHY LABORATORY

L	T	P	C
0	0	2	1

Pre-requisite Nil

Course Description:

This course offers hands-on experience with fundamental and modern cryptographic algorithms, enabling students to implement classical ciphers, block ciphers, public-key cryptography, and hash-based security mechanisms while understanding their internal workings, security strengths, weaknesses, and cryptanalysis, ultimately preparing learners to design, implement, and evaluate secure cryptographic solutions for real-world applications.

Course Objectives:

The objectives of this course are to

1. To understand and implement classical and modern cryptographic algorithms.
2. To gain practical skills in symmetric and asymmetric key operations.
3. To analyze secure key exchange mechanisms and identify vulnerabilities.
4. To explore various cryptographic modes, protocols, and secure random number generation.
5. To develop the ability to evaluate and apply cryptographic protocols in real-world security applications.

List of Experiments:

1. Write a program to implement classical cryptosystems such as Shift Cipher, Affine Cipher, and Vigenère Cipher, and perform basic cryptanalysis.
2. Write a program to implement substitution and transposition techniques for encrypting and decrypting messages.
3. Write a program to implement the DES algorithm, demonstrating key generation, Feistel rounds, and ciphertext production.
4. Write a program to simulate AES round operations including SubBytes, ShiftRows, MixColumns, and AddRoundKey.
5. Write a program to demonstrate block cipher modes of operation such as ECB, CBC, CFB, OFB, and CTR using any block cipher.
6. Write a program to implement the Diffie–Hellman Key Exchange and simulate a Man-in-the-Middle (MITM) attack.
7. Write a program to implement the RSA algorithm for key generation, encryption, decryption, and signing.
8. Write a program to generate cryptographically secure random numbers (CSPRNG) and compare them with pseudo-random number generators (PRNG).
9. Write a program to implement Elliptic Curve Cryptography (ECC) for key generation and encryption/decryption.
10. Write a program to compute message digests using SHA, generate HMAC, and verify digital signatures using RSA/ECDSA.
11. Implement secure email communication using Pretty Good Privacy (PGP). Generate public and private keys, encrypt and decrypt an email message, and apply digital signatures to ensure confidentiality, integrity, and authentication.
12. Implement secure email communication using S/MIME by generating X.509 certificates, configuring them in an email client, and sending digitally signed and encrypted email messages.

Course Outcomes:

After completion of the course, students will be able to

CO1: Apply classical and modern cryptographic techniques such as substitution, transposition, DES, AES, RSA, and ECC to secure data in various scenarios.

CO2: Demonstrate secure key generation and key exchange mechanisms including symmetric, asymmetric, and Diffie–Hellman protocols, and analyze their strengths and limitations.

CO3: Implement and evaluate cryptographic primitives and algorithms such as block cipher modes, hashing (SHA), HMAC, digital signatures, and secure random number generation.

CO4: Analyze vulnerabilities and attacks related to cryptographic systems such as brute-force, MITM, poor key management, and weak protocols, and propose security improvements.

CO5: Integrate cryptographic solutions into real-world security applications including authentication, secure communication, blockchain, SSL/TLS, and IoT security mechanisms..

Text Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice –8th Edition, Pearson Education,2023.
2. Behrouz A. Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, 3rd Edition, Mc Graw Hill Education, 2022.

References Books:

1. Rich Helton, Johennie Helton, Mastering Java Security: Cryptography Algorithms and Practices, John Wiley Publishers.
2. Charles P. Pleeger, “Security in Computing”, Pearson Education Asia, 5th Edition.
3. William Stallings, “Network Security Essentials: Applications and standards”, Person Education Asia.
4. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security: Private Communication in a public world”, Prentice Hall India, 6nd Edition. Hill, 2013

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

Skill Enhancement Courses

Skill Enhancement Course - I

25BACCSSC01 FUNCTIONAL PROGRAMMING

L	T	P	C
1	0	2	2

Prerequisite: Basic understanding of computers and programming concepts

Course Description:
This course introduces the fundamentals of Functional Programming and Python programming, covering programming paradigms, Python syntax, control structures, functions, data structures, strings, and file handling. It emphasizes problem-solving through simple real-world applications, enabling students to develop basic Python programs and apply computational thinking to practical programming tasks.

Course Objectives:
Students will be able to:

1. Introduce fundamental concepts of Functional Programming using Python.
2. Introduce Python programming fundamentals.
3. Understand basic programming constructs and functions.
4. Learn simple data structures and operations in Python.
5. Apply Python programming for String &file handling tasks.

UNIT I INTRODUCTION TO FUNCTIONAL PROGRAMMING 6 hours

Introduction
Programming Paradigms, Overview of Functional Programming, Features of Functional Programming, Functions vs Procedures, Expressions and Evaluation, Pure Functions, Side Effects, Immutability (basic idea), Benefits of Functional Programming.

Sample Problem Statements:

Pure Function for Academic Evaluation
A university grading system calculates whether a student has passed or failed based only on internal and external marks.
Design a **pure function** that takes marks as input and returns the result without modifying any external data.

Immutability in Student Records
A student record contains a list of registered courses. Instead of modifying the original list, the system creates a new updated list when a student adds a course.
Write a program to demonstrate the difference between **mutable and immutable data** in this scenario.

Expression-Based Result Calculation
An online examination portal decides a student's result using a single expression based on marks.
Create an **expression-based program** that outputs **Pass or Fail** without using multiple statements.

UNIT II INTRODUCTION TO PYTHON 6 hours

Introduction
Introduction to Python Programming, Features of Python, Python Syntax and Structure, Variables and Data Types, Input and Output Operations, Basic Operators.

Sample Problem Statements:

User Interaction for College Application
A college portal collects a student's name, roll number, and branch and displays them in a formatted manner.
Write a Python program using **input and output operations** to achieve this.
Basic Calculator for Fee Calculation
An admin wants to calculate total fees by performing addition, subtraction (scholarship), and multiplication (fine per day).
Develop a program that performs **simple arithmetic operations** based on user input.
Land Measurement System
A real-estate application calculates the area of a rectangular plot based on length and breadth entered by the user.
Write a Python program to compute and display the **area of the rectangle**.

UNIT III CONTROL STRUCTURES AND FUNCTIONS 6 hours

Control Structures
Conditional Statements (if, if-else), Looping Statements (for, while), Introduction to Functions, Function Definition and Calling, Parameters and Return Values.

Sample Problem Statements:
Even or Odd Seat Allocation
A transport system assigns seats differently for even and odd ticket numbers.
Create a program using **conditional statements** to check whether a given ticket number is even or odd.
Factorial for Combinatorics
A mathematics department application calculates factorial values for permutation and combination formulas.
Write a **function** that accepts a number and returns its factorial.
Multiplication Table Generator
An educational app helps students practice tables.
Develop a program using **loops** to generate the multiplication table for a given number.

UNIT IV DATA STRUCTURES IN PYTHON 6 hours

Data Structures
Basic Operations on Data Structures: Lists, Tuples, Sets, Dictionaries, Iterating through data structures.

Sample Problem Statements:
Student Marks Management Using Lists
A teacher stores marks of students in a list and performs operations like adding marks, removing incorrect entries, and displaying all marks.
Write a program to demonstrate **basic list operations**.
Immutable Exam Date Storage
Exam dates once finalized should not be changed.
Use a **tuple** to store exam dates and display them, highlighting immutability.
Student Information System Using Dictionary
A department stores student details such as roll number, name, and grade.
Create a program using a **dictionary** to store and display student information.

UNIT V STRINGS &FILE HANDLING 6 hours

Strings

String Creation, String Operations, String Functions

File Handling

Reading from Files, Writing to Files, Processing File Data, Simple Python-based Applications.

Sample Problem Statements:

Text Processing for Feedback System

An online feedback system analyzes student comments.
Write a program to perform **basic string operations** like length calculation, case conversion, and substring extraction.

Data Reversal for Security

A simple security application reverses a username before storing it.
Develop a program to **reverse a given string**.

Student Record File Management

A college stores student records in a text file.
Write a program to **write student data to a file and read it back**, displaying the contents neatly.

Course Outcomes:

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

CO1: Explain the basic functional programming concepts.

CO2: Identify and use fundamental Python programming elements such as variables, data types and operators.

CO3: Construct simple Python programs using control structures and user-defined functions.

CO4: Apply basic operations on Python data structures for organizing and processing data.

CO5: Apply String operations &file handling in Python programming.

Text Books:

1. Downey, A. B., *Think Python: How to Think Like a Computer Scientist*, 3rd Edition, O'Reilly Media, 2024.
2. Matthes, E., *Python Crash Course*, 3rd Edition, No Starch Press, 2023.
3. Severance, C. R., *Python for Everybody: Exploring Data Using Python 3*, 2024 Update.

Reference Books:

1. Dierbach, C., *Introduction to Computer Science Using Python: A Computational Problem-Solving Focus*, 2nd Edition, Wiley, 2015.
2. Ramalho, L., *Fluent Python*, 2nd Edition, O'Reilly Media, 2022.
3. Slatkin, B., *Effective Python: 125 Specific Ways to Write Better Python*, 3rd Edition, Addison-Wesley Professional, 2024.
4. Python Software Foundation, *Python Documentation (Python 3.13)*, 2025.

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

Skill Enhancement Course - II

25BACCSSC02 FULL STACK DEVELOPMENT I

L	T	P	C
1	0	2	2

Course Description:

This course provides hands-on training in web application development using HTML5, CSS, JavaScript, PHP, MongoDB covering the complete process from webpage design to dynamic, database-driven applications. Students learn to create structured web pages, apply styling and layouts, develop interactive client-side scripts, and perform efficient database operations. The course emphasizes practical implementation of forms, multimedia, validation, security, and server-side programming. By the end of the course, learners gain the skills required to design, develop, and manage full-stack web applications.

Course Objectives:

Students will be able to:

1. Make use of HTML elements and their attributes for designing static web pages
2. Build a web page by applying appropriate CSS styles to HTML elements
3. Experiment with JavaScript to develop dynamic web pages and validate forms
4. Build a web application using PHP
5. Understand the application using MongoDB connective.

UNIT I **6 hours**

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

[

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

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

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

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

UNIT II **6 hours**

HTML5 - Cascading Style Sheets, Types of CSS - Selector forms-5. CSS with Color, Background, Font, Text and CSS Box Model

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

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

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

d. Write a program to apply different types of selector forms

- Simple selector (element, id, class, group, universal)
- Combinator selector (descendant, child, adjacent sibling, general sibling)
- Pseudo-class selector
- Pseudo-element selector
- Attribute selector

e. Write a program to demonstrate the various ways you can reference a color in CSS.

f. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.

g. Write a program using the following terms related to CSS font and text:

i. font-size ii. font-weight iii. font-style

iv. text-decoration v. text-transformation vi. text-alignment

h. Write a program, to explain the importance of CSS Box model using

i.Content ii. Border iii. Margin iv. Padding

UNIT III **6 hours**

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

a. Write a program to embed internal and external JavaScript in a web page.

b. Write a program to explain the different ways for displaying output.

c. Write a program to explain the different ways for taking input.

d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not

e. Write a program using the following object properties and methods.

i. Document object
ii. array object
iii. math object
iv. string object
v. regex object
vi. date object

UNIT IV **6 hours**

Introduction to PHP covering features, syntax, embedding in HTML, data types, variables, constants, operators, control structures, and functions. HTML–PHP form integration with form controls, data handling using $_GET/$_POST, validation techniques, error handling, and display. Database fundamentals including design concepts, tables, keys, relationships, normalization, and PHP–MySQL connectivity using MySQLi/PDO

Application Development using PHP

a. Design a PHP web page using following form controls:
   i. Text box
   ii. Radio button
   iii. Check box
   iv. Buttons. Also validate the form data.

b. Write PHP program to
   i. Set cookies and read it
   ii. Demonstrate session management.

c. Develop a PHP web application to:
   i. Enter student data into a MySQL database
   ii. Retrieve and present data from the database.

d. Develop a PHP application with signup and login functionality. The application should be able to display employee information. Employees should be able to edit their information, except for their mobile number and name.

UNIT V APPLICATIONS **6 hours**

MongoDB fundamentals including databases, collections, documents, query language, installation, Mongo Shell, CRUD and aggregation operations, and Node.js driver for reading from and writing to MongoDB with schema initialization.

Modularization using Webpack covering back-end and front-end modules, transform and bundle process, library bundling, hot module replacement, debugging, Define Plugin build configuration, and production optimization.

a. Creating Database and Collection, create a database called company, create a collection named employees and display all collections

b. Node.js MongoDB Connection: Create a Node.js application, Connect Node.js to MongoDB using MongoDB driver, Display successful connection message

c. Backend Module (Node.js): Create a math module, Import and use it in main program

d. Frontend Module (ES6): Create JavaScript modules using export and import, Display output in browser console

Course Outcomes:

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

CO1: Apply HTML and CSS for Static Web Pages development.

CO2: Implement CSS styles to HTML elements code for web page.

CO3: Build dynamic Web Page Development with JavaScript.

CO4: Create dynamic web application using PHP.

CO5: Develop application using MongoDB connective.

Text Books:

1. Sebesta, R. W., Programming the World Wide Web, 9th Edition, Pearson Education, 2023.
2. Dean, J., Web Programming with HTML5, CSS, and JavaScript, 2nd Edition, Jones & Bartlett Learning, 2024.
3. Programming PHP: Creating Dynamic Web Pages, 4th Edition, Kevin Tatroe and Peter MacIntyre, 2020.
4. MongoDB Data Modeling and Schema Design Paperback, DanielCoupal, Pascal Desmarets, Steve Hoberman ,2023.

Reference Books:

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

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