ఆంధ్రప్రదేశ్ కేంద్రీయ విశ్వవిద్యాలయం आंध्रप्रदेश केंद्रीय विश्वविद्यालय

CENTRAL UNIVERSITY OF ANDHRA PRADESH

Jnana Seema, Ananthapuramu, India - 515701

DEPARTMENT OF COMPUTER SCIENCE & AI

SCHOOL OF INTERDISCIPLINARY AND APPLIED SCIENCES

Bachelors Programme

(as per the National Credit Framework 2023)

Vidya Dadati Vinayam

(Education Gives Humility)

B.Tech. Computer Science and Engineering

Programme Structure

(Effect from Academic Year 2025 - 26)

CONTENTS

1. Introduction to the Programme

The **B. Tech in Computer Science and Engineering (CSE)** at Central University of Andhra Pradesh is a forward-looking undergraduate programme to be introduced from the academic year 2025–26, developed in alignment with the **National Education Policy (NEP) 2020** and **All-India Council for Technical Education (AICTE)**. The programme is designed to establish a solid foundation in fundamental computing concepts while integrating modern technological advancements and interdisciplinary approaches. It offers flexible entry and exit options, ensuring an inclusive and industry-relevant learning experience.

The curriculum covers essential areas such as programming, algorithms, data structures, operating systems, databases, and computer networks. It also introduces advanced and emerging domains like Artificial Intelligence, Machine Learning, Data Science, Blockchain, Cybersecurity, Cloud Computing, and the Internet of Things (IoT). In addition, the programme emphasizes professional ethics, communication, teamwork, innovation, and sustainability.

Through structured coursework, hands-on lab sessions, and exposure to a wide range of subjects, students develop both theoretical understanding and practical competence. Upon completion, graduates will possess the technical skills, analytical mindset, and problem-solving abilities needed to pursue careers in software development, research, IT services, or higher education. The programme aspires to produce technically proficient and socially responsible engineers equipped to meet the demands of a rapidly evolving technological world

2. Programme Vision

The Vision of the Bachelor of Technology (B.Tech) Programme is to develop globally competent and pioneering engineering professionals who are innovative, skilled, and capable of delivering sustainable and impactful solutions to global challenges.

3. Programme Educational Objectives (PEOs)

PEO1: Develop the ability to formulate, analyze, and solve real-world computing problems using strong foundations in computer science, mathematics, and engineering principles.

PEO2: Pursue higher education or engage in research activities in computer science and related interdisciplinary fields, contributing to academic and technological advancements.

PEO3: Design, implement, test, and maintain reliable software systems using modern tools and technologies that address both industrial and societal needs.

PEO4: Exhibit professionalism through effective communication, teamwork, ethical conduct, leadership qualities, and entrepreneurial skills to work productively in diverse environments.

PEO5: Embrace lifelong learning, technological adaptability, and collaborative research that contribute to innovation, sustainable development, and nation-building efforts.

4. Programme Outcomes (POs)

The purpose of this programme is to equip students with advanced knowledge and practical skills in Computer Science and Engineering. After successful completion of the programme, students will be able to:

PO1 Engineering Knowledge Apply the knowledge of mathematics, science, engineering fundamentals, and computer science to solve complex engineering problems.

PO2 Problem Analysis Identify, formulate, research literature, and analyze complex computing problems using principles of mathematics and computer science.

PO3 Design/Development of Solutions Design solutions for complex engineering problems and systems that meet specified needs with appropriate consideration for public health, safety, culture, and environment.

PO4 Conduct Investigations of Complex Problems Use research-based knowledge and methods to analyze, interpret data, and derive valid conclusions.

PO5 Modern Tool Usage Create, select, and apply modern engineering and IT tools for modeling, simulation, and solving engineering problems.

PO6 The Engineer and Society Apply contextual knowledge to assess societal, legal, health, safety, and cultural issues relevant to engineering practice.

PO7 Environment and Sustainability Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate the need for sustainable development.

PO8 Ethics Apply ethical principles and commit to professional ethics and responsibilities in engineering practice.

PO9 Individual and Team Work Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10 Communication Communicate effectively on engineering activities with peers, the engineering community, and society at large through reports, documentation, and presentations.

PO11 Project Management and Finance Demonstrate knowledge of engineering and management principles and apply them to project work as a team member or leader.

PO12 Life long Learning Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning.

5. Programme-Specific Outcomes (PSOs):

The learning outcomes are designed to equip students with the knowledge, skills, and abilities necessary to succeed in careers related to economics, data analysis, and decision-making:

PSO1: Ability to design, develop, and maintain software systems using programming languages, data structures, algorithms, and software engineering practices.

PSO2: Ability to pursue higher studies in a multidisciplinary perspective leading to Masters and Research degrees.

PSO3: Ability to apply the Computer Science and Engineering concepts in different fields of Engineering culminating in successful careers and entrepreneurs with a focus on societal problem solving.

5. Pedagogy of the Programme

The B. Tech in Computer Science and Engineering curriculum emphasizes a blend of theoretical knowledge and practical application through a combination of structured coursework and hands-on laboratory components. The pedagogy is designed to foster a close interaction between students and teachers, allowing for individual attention. By integrating foundational and emerging technological concepts into real-world scenarios through hands-on activities and problem-solving labs, students are encouraged to engage in critical thinking and analysis. Learning outcomes are continuously monitored and achieved through a system of tests, assignments, and quizzes, preparing students to be technically proficient, innovative, and socially responsible engineers.

6. Programme Structure

- The Bachelor of Technology (B.Tech) in Computer Science and Engineering is a four-year undergraduate programme divided into eight semesters, comprising a total of 160 credits.
- The curriculum integrates Core Courses, Department Electives, Open Electives, Skill Development Courses, Laboratory Components, Internship, and Project/Dissertation Work, ensuring a comprehensive and industry-relevant learning experience.

  Semester-wise Structure:
  - **Semester I:** Foundational courses such as Mathematics, Engineering Physics, Basic Electrical and Electronics Engineering, Programming in C, and Communication Skills, along with laboratory components and audit courses like Sports and Yoga.
  - **Semester II:** Introduces courses in Mathematics-II, Digital Logic Design, Python Programming, Engineering Drawing, Design Thinking, and Human Values, supported by labs and workshop-based learning.
  - **Semester III:** Covers core computing subjects such as Data Structures, Object-Oriented Programming (Java), Computer Organization, and Discrete Mathematics, along with skill-based courses and laboratory practice.
  - **Semester IV:** Includes advanced core subjects like Operating Systems, Database Management Systems, Design and Analysis of Algorithms, Web Technologies, and Theory of Computation, supported by lab components and environmental studies.
  - **Semester V:** Focuses on specialized core areas such as Artificial Intelligence & Machine Learning, Computer Networks, Compiler Design, along with Department Elective-I, internship (SIP), and lab work.
  - **Semester VI:** Covers Software Engineering, Data Mining, Internet of Things, and Department Elective-II, along with lab components and soft skills training.
  - **Semester VII:** Includes advanced topics such as Cryptography and Network Security, Department Elective-III, Open Elective, Organizational Behaviour, and Project/Dissertation-I, along with labs.
  - **Semester VIII:** Dedicated to Project/Dissertation-II (major project) and MOOC-based Open Elective, enabling students to apply their knowledge to real-world problems and research.

B. Tech Computer Science and Engineering (CSE)

Department of Computer Science & AI

School of Interdisciplinary and Applied Sciences

Semester and Course wise Credits Distribution

Sem	Foundational Courses				Core Courses Credits				Skill Course & Elective Courses				Humanities				Lab Components			Audit Courses		SIP		Dissertation		Total
Sem	Code	T	L	Total	Code	T	L	Total	Code	T	L	Total	Code	T	L	Tot	Code	L	Total	Code	Total	Code	Total	Code	Total	Credits
I	*BTC101*	3	0	3	*BTC103*	3	0	3	—	—	—	—	BTC105	3	0	3	*BTC106*	2	2	BTC109	0	—	—	—	—	
	BTC102	3	0	3	*BTC104*	3	0	3									*BTC107*	2	2	—	—					
																	BTC108	2	2	—	—					
	Total	6	0	6	*Total*	6	0	6					Total	3	0	3	*Total*	6	6	Total	0					21
II	*BTC201*	3	0	3	*BTC202*	1	2	3	—	—	—	—	BTC205	2	0	2	*BTC207*	2	2	BTC209	0	—	—			
					BTC203	3	0	3					BTC206	2	0	2	*BTC208*	1	1							
					BTC204	3	1	4																		
	Total	3	0	3	*Total*	7	3	10					Total	4	0	4	*Total*	3	3	Total	0					20
III	*BTC301*	3	0	3	*BTC302*	3	1	4	BTC306	1	1	2	—		—	—	*BTC307*	2	2			—	—	—	—	
					BTC303	3	0	3									*BTC308*	2	2							
	BTC305	3	0	3	*BTC304*	3	0	3																		
	Total	6	0	6	*Total*	9	1	10	Total	1	1	2					*Total*	4	4							22
IV	—	—	—	—	*BTC401*	3	0	3	BTC406	1	1	2	—		—	—	*BTC407*	2	2	BTC409	0	—	—			
					BTC402	3	1	4									*BTC408*	2	2							
					BTC403	3	0	3																		
					BTC404	3	1	4																		
					BTC405	3	0	3																		
					Total	15	2	17	Total	1	1	2					*Total*	4	4							23
V	—	—	—	—	*BTC501*	3	0	3	*BTC504*	3	0	3	BTC505	3	0	3	*BTC506*	2	2	BTC510	0	BTC509	2	—		
					BTC502	3	0	3									*BTC507*	2	2							
					BTC503	3	0	3									*BTC508*	2	2							
					Total	9	0	9	*Total*	3	0	3	Total	3	0	3	*Total*	6	6	Total	0	Total	2			23
VI	—	—	—	—	*BTC601*	3	0	3	*BTC604*	3	0	3	BTC605	1	1	2	*BTC606*	2	2			—	—	—		
					BTC602	3	0	3									*BTC607*	2	2							
					BTC603	3	0	3									*BTC608*	2	2							
					Total	9	0	9	*Total*	3	0	3	*Total*	1	1	2	*Total*	6	6							20
VII	—	—	—	—	*BTC701*	3	0	3	*BTC702*	3	0	3	BTC703	3	0	3	*BTC705*	2	2			—	—	BTC707	4	
									BTC704	3	0	3					*BTC706*	2	2							
					Total	3	0	3	*Total*	6	0	6	Total	3	0	3	*Total*	4	4					Total	4	20
VIII	*BTC802*	3	0	3	—	—	—	—	—	—	—	—	—		—	—	—	—	—	—	—	—	—	BTC801	8	11
	Total	3	0	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Total	8	
Total Credits	—	18	0	18		58	6	64		14	2	16		14	1	15		33	33		0		2		12	160
Percentage (%)	—	—	—	11.2%	—	—	—	40%	—	—	—	10%	—		—	9.3%	—	—	20.6%			—	1.25%		7.5%	100%

Note: T: Theory **L/P:** Lab/Practicals **SIP:** Summer Internship Project

B. Tech Computer Science and Engineering (CSE)

Department of Computer Science & AI

School of Interdisciplinary and Applied Sciences

Programme Structure

Semester I						
Sl. No.	Course Code	Title of the Course	Credit Points	Credit Distribution		
				L*	T*	P*
1.	BTC101	Mathematics-I (Linear Algebra & Calculus)	3	3	0	0
2.	BTC102	Engineering Physics	3	3	0	0
3.	BTC103	Basic Electrical and Electronics Engineering	3	3	0	0
4.	BTC104	Problem Solving and Programming Using C	3	3	0	0
5.	BTC105	Communicative English	3	3	0	0
6.	BTC106	Engineering Physics Lab	2	0	0	2
7.	BTC107	Basic Electrical and Electronics Engineering Lab	2	0	0	2
8.	BTC108	Problem Solving and Programming Lab	2	0	0	2
9.	BTC109	Sports and Yoga (Audit Course)	0	2*	0	0
Total			**21**	**14**	**0**	**6**

Semester II						
Sl. No.	Course Code	Title of the Course	Credit Points	Credit Distribution		
				L*	T*	P*
1.	BTC201	Mathematics-II (Differential Equation and Vector Calculus)	3	3	0	0
2.	BTC202	Computer Aided Engineering Drawing	3	1	0	2
3.	BTC203	Digital Logic Design	3	3	0	0
4.	BTC204	Python Programming	4	3	0	1
5.	BTC205	Design Thinking	2	2	0	0
6.	BTC206	Universal Human Values	2	2	0	0
7.	BTC207	Digital Logic Design Lab	2	0	0	2
8.	BTC208	IT Workshop	1	0	0	1
9.	BTC209	NSS (Audit Course)	0	0	0	2*
Total			**20**	**14**	**0**	**6**

Semester III						
Sl. No.	Course Code	Title of the Course	Credit Points	Credit Distribution		
				L*	T*	P*
1.	BTC301	Mathematics-III (Probability & Statistics)	3	3	0	0
2.	BTC302	Objects Oriented Programming through Java	4	3	0	1
3.	BTC303	Data Structures	3	3	0	0
4.	BTC304	Computer Organization and Architecture	3	3	0	0
5.	BTC305	Discrete Mathematics	3	3	0	0
6.	BTC306	Introduction to R Programming (Skill Course -I)	2	1	0	1
7.	BTC307	Data Structures lab	2	0	0	2
8.	BTC308	Computer Organization and Architecture Lab	2	0	0	2
Total			**22**	**16**	**0**	**6**

Semester IV						
Sl. No.	Course Code	Title of the Course	Credit Points	Credit Distribution		
				L*	T*	P*
1.	BTC401	Design and Analysis of Algorithms	3	3	0	0
2.	BTC402	Operating Systems	4	3	0	1
3.	BTC403	Database Management Systems	3	3	0	0
4.	BTC404	Web Technologies	4	3	0	1
5.	BTC405	Theory of Computation	3	3	0	0
6.	BTC406	Power BI (Skill Course -2)	2	1	0	1
7.	BTC407	Design and Analysis of Algorithms Lab	2	0	0	2
8.	BTC408	Database Management Systems Lab	2	0	0	2
9.	BTC409	Environmental Sciences	0	2*	1*	0
Total			**23**	**16**	**0**	**7**

Semester V						
Sl. No.	Course Code	Title of the Course	Credit Points	Credit Distribution		
				L*	T*	P*
1.	BTC501	Artificial Intelligence and Machine Learning	3	3	0	0
2.	BTC502	Computer Networks	3	3	0	0
3.	BTC503	Compiler Design	3	3	0	0
4.	Department Elective-I: Any one of the following		3	3	0	0
	BTC504	A. Information Retrieval B. Cloud Computing C. Advanced Data Structures D. Human Computer Interaction E. Mobile Computing F. Software Testing G. Computer Graphics				
5.	BTC505	Managerial Economics and Financial Analysis	3	3	0	0
6.	BTC506	Artificial Intelligence and Machine Learning Lab	2	0	0	2
7.	BTC507	Computer Networks Lab	2	0	0	2
8.	BTC508	Compiler Design Lab	2	0	0	2
9.	BTC509	Summer Internship Project	2	0	0	2
10.	BTC510	Constitution of India	0	2*	1*	0
Total			**23**	**15**	**0**	**8**

Semester VI						
Sl. No.	Course Code	Title of the Course	Credit Points	Credit Distribution		
				L*	T*	P*
1.	BTC601	Software Engineering	3	3	0	0
2.	BTC602	Data Mining	3	3	0	0
3.	BTC603	Internet of Things	3	3	0	0
4.	Department Elective-II: Any one of the following		3	3	0	0
	BTC604	A. Deep Learning B. Distributed Systems C. Embedded Systems D. Cyber Security E. Natural Language Processing F. Parallel Computing G. Robotics H. Game Development				
5.	BTC605	Soft Skills	2	1	0	1
6.	BTC606	Software Engineering Lab	2	0	0	2
7.	BTC607	Data Mining Lab	2	0	0	2
8.	BTC608	Internet of Things Lab	2	0	0	2
Total			**20**	**13**	**0**	**7**

Semester VII						
Sl. No.	**Course Code**	**Title of the Course**	**Credit Points**	**Credit Distribution**		
				L*	**T***	**P***
1.	BTC701	Cryptography and Network Security	3	3	0	0
2.	BTC702	**Department Elective-III: Any one of the following** A. Image Processing and Computer Vision B. Deep Computer Vision C. Quantum Computing D. Virtual Reality E. Data Analytics F. Advanced Compiler Design G. Digital Signal Processing H. Pattern Recognition	3	3	0	0
3.	BTC703	Organizational Behaviour	3	3	0	0
4.	BTC704	Open Elective I through MOOCs	3	3	0	0
5.	BTC705	Cryptography and Network Security Lab	2	0	0	2
6.	BTC706	Department Elective- III Lab	2	0	0	2
7.	BTC707	Dissertation-I	4	1	0	3
		Total	**20**	**13**	**0**	**7**

Semester VIII						
Sl. No.	**Course Code**	**Title of the Course**	**Credit Points**	**Credit Distribution**		
				L*	**T***	**P***
1.	BTC801	Dissertation -2	8	0	0	8
2.	BTC802	Open Elective II through MOOCs	3	3	0	0
		Total	**11**	**3**	**0**	**8**
		Grand Total	**160**	**104**	**0**	**55**

Note: **L:** Lectures; **T:** Tutorials **P:** Practical /Lab

Semester-wise Credit Distribution

Semester	Total Credits	Cumulative credit at the end of the semester
I	21	21
II	20	41
III	22	63
IV	23	86
V	23	109
VI	20	129
VII	20	149
VIII	11	160

Note:

1. MOOCs are chosen by the student based on the availability of the courses offered on SWAYAM platform.
2. The Programme template and the title of the courses are tentative, any changes as required may be made.

I. Credit Requirement

- The minimum number of credits required to be earned by a student for the award of the degree is **160**.

II. Assessment Pattern

- **Theory Courses**:
    - **Internal (40%)** – Formative evaluation comprising two best out of three tests (each test carrying a maximum of 15 marks, totalling 30 marks), along with seminar/assignments (10 marks).
    - **External (60%)** – Summative evaluation through the semester-end examination.

  End-Semester Examination:
    - Maximum Marks: **60**
    - Duration: **3 Hours**
- **Theory with Laboratory Courses:**
    - **Internal (60%)** – Formative evaluation comprising two best out of three tests (each test carrying a maximum of 15 marks, totalling 30 marks),Practical lab (20 marks) and seminar/assignments (10 marks).
    - **External (40%)** – Summative evaluation through the semester-end examination.

  End-Semester Examination:
    - Maximum Marks: **40**
    - Duration: **2 Hours**

III. Laboratory Components:

- **Internal (60%)** – Continuous assessment through lab work and internal examination.
- **External (40%)** – Summative evaluation through the semester-end practical examination.

IV. Dissertation or Project Report:

- **Maximum Marks: 100**
    - Dissertation Evaluation - 60 Marks

- Seminar and Viva-Voce- 40 Marks

B. Tech Computer Science and Engineering (CSE)

Department of Computer Science & AI

School of Interdisciplinary and Applied Sciences

Important Information to Students

i. Programme: B. Tech Computer Science and Engineering

ii. Eligibility: 10+2 level of education (Intermediate/ PUC/ CBSE/ ICSE/ HSC or an equivalent examination) with at least 60% marks (55% for SC/ST candidates) in aggregate in three subjects: Physics, Mathematics, and any one subject from Chemistry/Computer Science/Electronics/Information Technology/ Biology/Informatics Practices/Biotechnology/Technical Vocational subject/Agriculture/Engineering Graphics/ Business Studies/Entrepreneurship

iii. The minimum duration for completion of any UG Programme is eight semesters (four academic years) and the maximum duration is sixteen semesters (8 academic years) or as per amendments made by the regulatory bodies from time to time.

iv. A student should have minimum 75% attendance in classes, seminars, practical/ lab in each course of study without which he/she will not be allowed for the Semester -end examination.

v. All theory courses in the programme shall have Continuous Internal Assessment (CIA) component of 40 marks and a Semester-end component for 60 marks. The minimum pass marks for a course is 50%.

vi. In case of courses with lab component Continuous Internal Assessment (CIA) component shall be of 60 marks and Semester-End Component for 40 marks. The minimum pass marks for a course is 50%.

vii. The student is given three Continuous Internal Assessment (CIA) tests per semester in each course from which the best two performances are considered for the purpose of calculating the marks in CIA. A record of the continuous assessment is maintained by the academic unit. The three internal tests are conducted for 15 Marks each, out of the best two tests' scores are considered for 30 marks and for Academic Participation (10 marks)

viii. The remaining 10 marks are awarded based on participation and performance in:

  a. Assignments
  b. Class presentations
  c. Seminars
  d. Quizzes

ix. A student should pass separately in both CIA and the Semester-end Examination (SEE), i.e., a student should secure 20 (50% of 40) out of 40 marks for theory in CIA and 30 (50% of 60) out of 60 marks for theory in the SEE.

x. Semester-end examination shall consist of Objective type questions, descriptive type questions, short answer questions and case studies or any others.

xi. A student failing to secure the minimum pass marks in the CIA is not allowed to take the Semester-end examination of that course. She/he has to redo the course by attending special classes for that course and get the pass percentage in the internal tests to become eligible to take the semester-end examination.

xii. Students failing a course due to lack of attendance should redo the course.

xiii. Re-evaluation is applicable only for theory papers and shall not be entertained for other components such as practical/ thesis/ dissertation/ internship etc.

xiv. An on-campus elective course is offered only if 30 or 50% of the students registered, whichever is higher.

SEMESTER WISE
DETAILED SYLLABI

SEMESTER-I

Course Code : **BTC101** Course Type : **Basic Science** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Mathematics-I (Linear Algebra & Calculus)**

Course Objectives:

- To understand the basic foundation knowledge on linear algebra and calculus essential for solving engineering problems.
- To develop analytical skills through the application of differentiation, integration, and matrix operations.
- To apply the concepts of Engineering Mathematics in solving the real-world applications.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Apply fundamental concepts of calculus, determine the Rank of a matrix using echelon and normal forms and understand its implications for solving linear systems.
- **CO2:** Understand and manipulate vector spaces by exploring subspaces, linear independence, basis, dimension, and linear transformations
- **CO3:** Analyse multivariable functions, Jacobians, optimization techniques such as Lagrange multipliers, vector spaces and perform linear transformations using basis and dimension concepts.
- **CO4:** Solve systems of linear equations using matrix methods, and interpret matrix properties such as rank, eigenvalues, and diagonalizability
- **CO5:** Evaluate definite and multiple integrals for solving problems related to area, volume, and special functions

COs /POs Mapping:

Cos/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	2	2	-	-	-	-	-	-	-
CO2	3	3	-	2	2	-	-	-	-	-	-	-
CO3	3	3	2	3	2	-	-	-	-	-	-	-
CO4	3	3	-	3	3	-	-	-	-	-	-	-
CO5	3	2	2	3	-	-	-	-	-	-	-	-

Course Outline:

Unit-I: **[08 Hours]**

Single Variable Calculus: Limits, continuity, rules of differentiation, Taylor's series, and its consequences, Indeterminate forms, L' Hospital's rule, mean value theorems: Rolle's theorem, Lagrange's mean value theorem, Cauchy's mean value theorem

Unit-II: **[08 Hours]**

Multivariable Calculus: Functions of several variables, partial derivatives, chain rules, total derivatives, Taylor and Maclaurin series in two variables, Jacobian, maxima and minima of functions of two variables, method of Lagrange multipliers.

Unit-III: **[10 Hours]**

Integral Calculus: Length of the curves, surface area of revolution, volume of solids of revolution, Leibnitz rule, improper integrals, beta and gamma functions. *Double and Triple Integrals:* evaluation of double, triple, surface, and volume integrals.

Unit-IV: **[08 Hours]**

Linear Algebra: Vector spaces, subspaces, linear independence and dependence of vectors, linear span, basis and dimension of vector spaces, linear transformation, and its properties

Unit-V: **[11 Hours]**

Matrix Theory: Rank of a Matrix using Echelon method, consistency of the system of linear equations in terms of matrix notions, eigenvalues and eigenvectors of a matrix, Caley-Hamilton theorem, similar matrices, diagonalizability, special matrices (symmetric, skew-symmetric, Hermitian, skew-Hermitian, orthogonal, and unitary matrices) and their properties.

Suggested Readings:

E. Kreyszig, "Advanced Engineering Mathematics", John Wiley & Sons, 10th ed., 2023.
J. Stewart, D.K. Clegg and S. Watson, "Calculus", Cengage Learning India, 9th ed., 2023.

References:

G. Strang, "Linear Algebra and its Applications", Brooks/Cole, 4th ed., 2006.
G.B. Thomas, M. D. Weir, and J.R. Hass, "Calculus", Pearson Education, 15th ed., 2024.
N. Piskunov, "Differential Calculus and Integral Calculus–I, II", CBS Publishers & Distributors, 1st ed., 1996.

Course Code : **BTC102** Course Type : **Basic Science** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Engineering Physics**

Course Objectives:

- To understand the principles, working, and applications of lasers and fiber optics.
- To comprehend the foundational concepts of quantum mechanics and their implications in physical systems.
- To explore the physical properties of nuclear structure, radioactivity, and elementary particle physics materials,
- including electrical, magnetic, and superconducting behaviour.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Understand the basic working and uses of lasers and optical fibers, electrical, magnetic, super conducting properties of materials.
- **CO2**: Explain the key ideas and experiments that led to quantum mechanics.
- **CO3**: Solve simple quantum problems using Schrödinger's equation.
- **CO4**: Describe the structure of the nucleus, radioactivity, and nuclear reactions.
- **CO5**: Identify basic particles and understand fundamental forces in nature.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	-	-	-	-	-	-	-	-
CO2	3	2	-	-	-	-	-	-	-	-	-	-
CO3	3	2	-	2	-	-	-	-	-	-	-	-
CO4	3	-	-	-	-	-	-	-	-	-	-	-
CO5	3	-	-	-	-	-	-	-	-	-	-	-

Course Outline:

Unit-I: **[09 Hours]**

Lasers: Introduction to lasers, Characteristics of lasers, Spontaneous and stimulated emissions, Einstein's coefficients, Population inversion and lasing action, He-Ne laser, Semiconductor laser, Applications of lasers.
Fiber Optics: Fermat's principle, Principle and construction of optical fibers, Acceptance cone, Numerical aperture, Types of fibers, Principle of fiber optic communication and Fiber optic sensors

Unit-II: **[09 Hours]**

Inadequacy of classical mechanics, Black body radiation, Planck's law and Planck's equation. Photoelectric effect and its features, Einstein's explanation, Wave-particle duality of radiation, de Broglie concept of matter waves, Electron diffraction-Davission and Germer experiment, Heisenberg's uncertainty principle.
Schrödinger's time-independent wave equation; wave function-physical significance in terms of probability density, Eigenvalues and Eigen functions, Application of Schrodinger wave equation for a Particle confined in one-dimensional infinite square well potential.

Unit-III: **[09 Hours]**

Nuclear properties and nuclear forces, nuclear models (qualitative): Liquid drop model, Shell model, nuclear reactions, Radioactivity – types of radiation, decay equation and half-life, applications of radioactivity. Fundamental forces in nature, Basics of particle physics, Classification of elementary particles-standard model of particle physics, Quark model.

Unit-IV: **[10 Hours]**

Physics of Materials: Dielectric materials – Polar and non-polar materials, electric polarization - Clausius-Mossotti Relation Conductors: classical free electron theory (Lorentz –Drude theory) – electrical conductivity, Quantum free electron theory (qualitative) Magnetic materials Diamagnetic, paramagnetic and ferromagnetic-properties and applications. Superconductors: definition – Meissner effect – type I & II superconductors – BCS theory (qualitative)

Unit-V: **[12 Hours]**

Origin of Nanotechnology, Nano Scale, Surface to Volume Ratio, Fabrication: Bottom-up approach Sol-gel Process and Chemical vapour deposition; Top-down approach –Ball milling and Lithography. Characterization techniques-SEM, EDAX. Physical, Chemical and Optical properties of Nano materials, Applications.

Suggested Readings:

W. T. Silfvast, "Laser Fundamentals," 2nd ed. New York, NY, USA: Cambridge University Press, 2004.
D. Halliday, R. Resnick, and J. Walker, "Fundamentals of Physics", 6th ed. New York, NY, USA: John Wiley & Sons, 2001.
A. Beiser, "Concepts of Modern Physics". New Delhi, India: Tata McGraw-Hill, 2010.

References:

R. Shankar, "Fundamentals of Physics II" New Haven, CT, USA, and London, UK: Yale University Press,2016.
C. Kittel, "Introduction to Solid State Physics", 8th ed. Hoboken, NJ, USA: John Wiley & Sons, 2005.
Richard Booker and Eary Boyseen, "Nanotechnology" John Wiley & sons, 2005
Bharat Bhushan. Encyclopedia of Nanotechnology" Springer Volume-1
Amretashis Sengupta and Chandan Kumar Sarkar, "Introduction to Nano: basics to nanoscience and nanotechnology"

Course Code : **BTC103** Course Type : **Core Course** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Basic Electrical and Electronics Engineering**

Course Objectives:

- To learn the basic concepts of fundamental laws and theorems of electrical circuits, electronic devices.
- To acquire the skills for design, develop the operating principles of electrical machines.
- To familiarize with sensors, transducers, and measurement instruments.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Analyze basic DC and AC electrical circuits using laws and theorems.
- **CO2**: Understand the working and characteristics of electrical machines like transformers, DC and AC
- motors.
- **CO3**: Explain the behaviour and application of diodes, transistors, and power electronic devices.
- **CO4**: Identify different types of sensors and transducers used in electrical measurements.
- **CO5**: Use electrical measuring instruments such as DMM and CRO for basic testing and analysis.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	1	-	-	-	-	-	-	-
CO2	3	2	3	2	1	-	-	-	-	-	-	-
CO3	3	2	2	2	2	-	-	-	-	-	-	-
CO4	2	2	2	2	1	-	-	-	-	-	-	-
CO5	3	2	2	2	2	-	-	-	-	-	-	-

Course Outline:

Unit-I: **[09 Hours]**

DC and AC Circuits: Kirchhoff's Voltage and Current Laws, Superposition Theorem, Star-Delta Transformations. Complex representation of Impedance, Phasor diagrams, Power & Power Factor, Solution of 1-Phase Series & Parallel Circuits.

Unit-II: **[09 Hours]**

Single Phase Transformers: Principle of Operation, EMF Equation, Phasor Diagram, Equivalent Circuit, Determination of Parameters, Regulation & Efficiency-DC Machines: Principle of Operation, Classification, EMF and Torque Equations, Characteristics, Speed Control Methods.

Unit-III: **[10 Hours]**

AC Machines: 3-Phase Induction Motor – Operation, Torque-Speed Characteristics & Applications; Alternator – Principle, EMF Equation. Electronic devices & Circuits: P-type and N-Type semiconductors, P-N junction diode and its I-V characteristics, Half-wave, and Full-wave rectifiers.

Unit-IV: **[08 Hours]**

Bipolar Junction Transistor – Operation and configurations (CE, CC, CB), Static Characteristics. SCR, MOSFET, IGBT – Basic operation-Sensors & Transducers: Thermocouple, Thermistor, RTD, Hall Effect, Piezoelectric

Unit-V: **[09 Hours]**

Measuring Instruments: Moving Coil & Moving Iron Ammeters and Voltmeters, Wattmeter (qualitative). Electronics Measurements: Digital Multimeter – Principle of Operation, Cathode Ray Oscilloscope (CRO) – basic use.

Suggested Readings:

E. Hughes, "Electrical & Electronic Technology", Pearson Education, 13th ed., 2021.
V. Del Toro, "Electrical Engineering Fundamentals", Pearson Education, 2nd ed., 2015.
V. K. Mehta, "Principles of Electrical & Electronics Engineering," S. Chand Publications, New Delhi, 3rd ed., 2010.
V. N. Mittle and A. Mittal, "Basic Electrical Engineering," Tata McGraw Hill, 2nd ed., 2005.

References:

J. Millman and C. Halkias, "Integrated Electronics: Analog and Digital Circuits and Systems, Tata McGraw-Hill Education," 2nd ed., 2010.
U. Bakshi and A. Bakshi, "Basic Electrical Engineering," Technical Publications, 2019.
A. E. Fitzgerald, C. Kingsley, and S. D. Umans, "Electric Machinery," McGraw-Hill Education, 7th ed., 2014.
S. J. Chapman, "Electric Machinery," McGraw Hill International Edition, 4th ed., 2017.

Course Code : **BTC104** Course Type : **Core Course** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Problem Solving and Programming Using C**

Course Objectives:

- To understand the basic concepts of structure, operation of computers and programming languages.
- To acquire skills for developing algorithms, problem-solving techniques.
- To enable students to handle advanced features in C such as dynamic memory allocation, structures, and file operations.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Explain the components of a computer system and classify different programming languages.
- **CO2:** Design algorithms and flowcharts to solve computational problems systematically.
- **CO3**: Develop C programs using variables, expressions, control structures, and loops.
- **CO4**: Implement modular programming techniques using functions, arrays, and parameter passing.
- **CO5:** Apply advanced C concepts like dynamic memory allocation, structures, and file I/O in practical applications.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	–	–	–	–	–	–	–	2
CO2	2	3	2	1	–	–	–	–	–	–	–	2
CO3	3	3	3	2	2	–	–	–	–	–	–	2
CO4	3	3	3	2	3	–	–	–	1	–	–	2
CO5	3	2	2	2	3	–	–	–	–	–	–	2

Course Outline:

Unit-I: **[10 Hours]**

Fundamentals of Computers - Components of a computer, Problems, Flowcharts, Memory, Variables, Values, Instructions, Programs, programming languages, concept of high-level, assembly and low-level programming languages,

Problem solving techniques – Algorithmic approach, characteristics of algorithm, Flowcharts, Problem solving strategies: Top-down approach, Bottom-up approach, Number systems and data representation.

Unit-II: **[10 Hours]**

Introduction to C- C character set- Identifiers and Keywords- Data types- Constants- Variables Declarations- Expressions- Statements- Symbolic Constants- Operators- Library Functions Data input and output: Single character input and output- Entering input data- Writing output data- gets and puts functions – Control Statements- Branching: if-else-looping: while- do..while for; Nested control Structures- switch statements- Break statements- Continue Statements, Comma operator- goto statements

Unit-III: **[09 Hours]**

Modular Programming- Functions and Procedures - Examples- Parameters passing methods - Arrays- Defining an array- Processing an array- Multi dimensional arrays- Pointers- Variables definitions and initializations- Pointer operators- Pointer expressions and arithmetic- Pointers and one-dimensional arrays - String operations.

Unit-IV: **[08 Hours]**

Functions- Defining function- Accessing a function- Function prototypes- Passing arguments to a function- Passing arrays to a function- Passing Pointers to function- Recursion – Dynamic memory allocation - malloc, calloc, realloc.

Unit-V: **[08 Hours]**

Structures – Declaration – Structures and Functions – Arrays of Structures – Pointers to structures – Typedef - Unions – Bit-fields - Files – Input / Output using files – fread, fwrite, fprintf, fscanf – Formatted input – File access - argc, argv

Suggested Readings:

B. S. Gottfried, Schaum's, "Outline of Programming with C," New York, NY, USA: McGraw-Hill, 3rd ed., 2017.

R. Thareja, "Computer Fundamentals and Programming in C," Oxford University Press, 3rd ed., 2023.

B. W. Kernighan and D. M. Ritchie, “The C Programming Language,” Pearson Education, 2nd ed., 1988.
V. Rajaraman, “Computer Basics and C Programming,” PHI Learning Pvt. Limited, 2006.

References:

E. Balagurusamy, “Computer Concepts and Programming in C,” McGraw Hill Education, 2nd ed., 2017.
J. R. Hanly and E. B. Koffman, “Problem Solving and Program Design in C,” Pearson Education, 8th ed.,2015.
Y. Kanetkar, “Let Us C,” BPB Publications, 21st ed., 2023.
H. M. Deitel and P. J. Deitel, “C: How to Program,” Pearson Education, 8th ed., 2015.
S. Anami, N. Angadi, and S. Manvi, “Computer Concepts and Programming,” PHI Learning Pvt. Ltd., 2nd ed., 2010.
R. S. Salaria, “Problem Solving and Programming in C,” Khanna Publishing House 5th ed., 2016.

Course Code : **BTC105** Course Type : **Humanities** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Communicative English**

Course Objectives:

- To learn basic concepts of speaking, writing English clearly in technical and academic settings.
- To acquire skills on build confidence in public speaking, teamwork, and professional discussions.
- To guide students on how to write emails, reports, resumes, and business messages.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Gain Knowledge on Speak and write English clearly and correctly.
- **CO2**: Analyze and prepare good reports, emails, résumés, and presentations.
- **CO3:** Design for learning under uncertainty building suitable teamwork and leadership.
- **CO4:** Acquire skills for effective Speaking, writing well in job interviews and startup pitches
- **CO5:** Develop effective language learning systems.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	–	–	–	–	–	–	–	2	3	–	2
CO2	2	–	3	–	2	–	–	–	2	3	2	2
CO3	–	–	–	–	–	2	–	2	3	3	2	2
CO4	2	-	-	-	-	-	-	-	-	3	-	-
CO5	2	-	-	-	-	-	-	-	-	3	-	2

Course Outline:

Unit-I: [09 Hours]

Functional grammar for technical communication, Vocabulary development with domain focus (technical jargon, academic register), Sentence construction, coherence and cohesion, Listening comprehension (TED Talks, AI/CS podcasts), Speaking practice: self-introduction, articulation, fluency, pronunciation.

Unit-II: [09 Hours]

Public speaking: structuring and delivering technical presentations, Group discussion and collaborative, communication, Cross-cultural and interdisciplinary communication, Communication ethics in AI: bias, transparency, responsibility, Interview skills and real-world scenarios

Unit-III: [09 Hours]

Email etiquette and official communication, Writing instructions and process descriptions (algorithms, procedures), Academic writing: paragraphing, referencing, summarizing, Report writing: lab/project reports, incident reports, Resume and Statement of Purpose writing, Crafting clear and compelling elevator pitches for tech products or AI solutions, Writing concise startup concept notes, executive summaries, and business emails.

Unit-IV: [09 Hours]

Introduction to Technical and Digital Writing, Differences between technical and general writing, Principles of Clear and Concise Writing, Organizing information logically, Writing effective proposals, Social Media and Digital Content Adapting technical content for social media Microblogging, posts, and threads- Document, Design and Structure, Avoiding jargon and ambiguity, Digital Writing Tools and Software Word processors, collaboration tools Introduction to Markdown, HTML basics.

Unit-V: [09 Hours]

Corporate communication vs. marketing communication, Visual and verbal corporate identity, Communication strategies for different audiences, Traditional media, Digital media and social platforms, Press releases and press conferences, Social media management, Crisis management principles Preparing and responding to crises, Delivering persuasive oral pitches to potential investors, partners, or incubators.

Suggested Readings:

Meenakshi Raman & Sangeeta Sharma. *Technical Communication: Principles and Practice*. Oxford University Press.

K. R. Lakshminarayanan. *English for Technical Communication*, Cambridge University Press

Course Code : **BTC106** Course Type : **Basic Science Lab** No. of Credits : **2.00** No. of Hours : **60**	**Course Title** **Engineering Physics Lab**

Course Objectives:

- To provide foundational understanding of laser physics, fiber optics, quantum mechanics, nuclear and particle physics, and materials science.
- To enable students to apply physical principles to engineering problems and modern technological applications.
- To develop attitude for independent and continues learning of latest concepts of Engineering Physics.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Explain the principles and applications of lasers and optical fibers in modern technology.
- **CO2:** Apply the concepts of quantum mechanics to understand the behaviour of microscopic particles.
- **CO3:** Describe nuclear properties, models, and basic ideas of particle physics and fundamental forces.
- **CO4:** Analyze the properties of dielectric, magnetic, conducting, and superconducting materials.
- **CO5:** Utilize the understanding of modern physics in the design and functioning of electronic and photonic devices.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	2	-	-	-	1	-	-	1
CO2	3	3	3	2	2	-	-	-	1	-	-	1
CO3	3	2	2	2	1	1	-	-	1	-	-	1
CO4	3	2	3	2	2	-	-	-	1	-	-	1
CO5	2	2	2	3	3	-	-	-	1	-	-	1

List of Experiments:

1. Study of Laser: Determination of Wavelength using Diffraction Grating / single slit / Double slit
2. Measurement of Numerical Aperture of an Optical Fiber
3. Determination of Planck's Constant using LED
4. Verification of Photoelectric Effect and Stopping Potential
5. Study of He-Ne Laser Beam Profile and Beam Divergence
6. Determination of Wavelength of Light using Newton's Rings
7. Determination of Energy Band Gap of a Semiconductor
8. Diffraction of Electron Beam (Davisson-Germer Experiment Simulation)
9. Determination of e/m by Thomson's Method
10. Measurement of Magnetic Susceptibility of a Material
11. B-H Curve Tracing using CRO (Magnetic Hysteresis Loop)
12. Determination of Dielectric Constant of a Material (liquid or solid)
13. Study of Hall Effect in Semiconductors
14. Superconductivity Demonstration (Meissner Effect, where available)
15. Verification of Heisenberg's Uncertainty Principle (using laser diffraction)
16. Four probe method: electrical resistance and resistivity

Suggested Readings:

W. T. Silfvast, "Laser Fundamentals," New York, NY, USA: Cambridge University Press, 2nd ed., 2004.
D. Halliday, R. Resnick, and J. Walker, "Fundamentals of Physics,", New York, NY, USA: John Wiley & Sons, 12th ed., 2021.
A. Beiser, S. Mahajan, and S. Rai Choudhury, "Concepts of Modern Physics," McGraw Hill Education, 8th ed., 2024.
R. Shankar, "Fundamentals of Physics II: Electromagnetism, Optics, and Quantum Mechanics," New Haven, CT, USA, and London, UK: Yale University Press, Expanded ed., 2020.
5. C. Kittel, "Introduction to Solid State Physics," Hoboken, NJ, USA: John Wiley & Sons, 8th ed., 2005

Course Code : **BTC107** Course Type : **Core Lab** No. of Credits : **2.00** No. of Hours : **60**	**Course Title** **Basic Electrical and Electronics Engineering Lab**

Course Objectives:

- Understand the basic principles of electrical and electronic circuits through practical
- experiments.
- Analyze and interpret the behaviour of transformers, DC machines, and AC machines under different operating conditions.
- Study the characteristics and operation of electronic components like diodes, BJTs, SCRs, and sensors, electrical and electronic instruments like Multimeter, CROs, and potentiometers, and gain hands-on experience to correlate theoretical concepts with real-world applications**.**

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Verify fundamental electrical laws and transformer characteristics through experiments.
- **CO2**: Conduct tests on DC motors and induction motors to evaluate their performance.
- **CO3:** Analyze the V-I characteristics of diodes and transistors.
- **CO4:** Acquire skills on CROs, Multimeter, and other instruments to measure and analyze electrical and electronic parameters.
- **CO5**: Explore the working of common sensors and transducers in real-time applications.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	2	-	-	-	-	-	-	-
CO2	3	2	3	2	2	-	-	-	-	-	-	-
CO3	3	2	2	1	3	-	-	-	-	-	-	-
CO4	2	2	3	2	3	-	-	-	-	-	-	-
CO5	2	2	2	3	3	-	-	-	-	-	-	-

List of Experiments:

1. Open Circuit and Short Circuit Tests on Single-Phase Transformer
2. Load Test on Single-Phase Transformer
3. Speed Control of DC Shunt Motor
4. Torque-Speed Characteristics of 3-Phase Induction Motor (Demo/Simulation)
5. Measurement of Resistance using Wheatstone Bridge
6. Calibration of Voltmeter using Potentiometer
7. Measurement of Signal Parameters using Oscilloscope
8. V-I Characteristics of a P-N Junction Diode
9. Full-Wave Rectifier with and without Filter
10. Input and Output Characteristics of BJT (CE Configuration)
11. VI Characteristics of SCR / MOSFET
12. Study of Thermistor / RTD Characteristics

Suggested Readings:

E. Hughes, “Electrical & Electronic Technology”, Pearson Education, 13th ed., 2021.
V. Del Toro, “Electrical Engineering Fundamentals”, Pearson Education, 2nd ed., 2015.
V. K. Mehta, “Principles of Electrical & Electronics Engineering,” S. Chand Publications, New Delhi, 3rd ed., 2010.
V. N. Mittle and A. Mittal, “Basic Electrical Engineering,” Tata McGraw Hill, 2nd ed., 2005.
J. Millman and C. Halkias, “Integrated Electronics: Analog and Digital Circuits and Systems, Tata McGraw-Hill Education,” 2nd ed., 2010.
U. Bakshi and A. Bakshi, “Basic Electrical Engineering,” Technical Publications, 2019.
A. E. Fitzgerald, C. Kingsley, and S. D. Umans, “Electric Machinery,” McGraw-Hill Education, 7th ed., 2014.
S. J. Chapman, “Electric Machinery,” McGraw Hill International Edition, 4th ed., 2017.
P. S. Bimbhra, “Electrical Machinery: Theory, Performance & Applications,” Khanna Publishers, 8th ed., 2021.

Course Code : **BTC108** Course Type : **Core Lab** No. of Credits : **2.00** No. of Hours : **60**	**Course Title** **Problem Solving and Programming using C Lab**

Course Objectives:

- Develop programming skills through a foundational course in C programming.
- Introduce basic decision-making and logical thinking via hands-on exercises.
- Inculcate algorithmic and programmatic thinking through practical coding tasks.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Demonstrate fundamental programming skills using the C language.
- **CO2**: Apply conditional and looping constructs to solve computational problems.
- **CO3**: Implement recursive and iterative solutions in programming tasks.
- **CO4**: Participate effectively in coding exercises and competitions using structured logic.
- **CO5**: Design and develop logical solutions for real-life computational problems using C.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	–	–	–	–	–	–	–	–	2
CO2	3	3	2	–	–	–	–	–	–	–	–	2
CO3	3	3	2	–	–	–	–	–	–	–	–	2
CO4	2	3	2	–	–	–	–	–	1	–	–	3
CO5	3	3	3	2	2	–	–	–	1	–	–	3

List of Experiments:

1. Develop a C program to test whether a number is prime using a loop construct.
2. Develop C programs to demonstrate the use of relational, arithmetic, logical, bitwise operators, and ternary operator.
3. Create a C program to compute factorial of a number using a loop and use break and continue where needed.
4. Write a C program to generate Fibonacci numbers using a loop.
5. Write a C program to print the following pattern:
   ```
   *
   **
   ***
   ****
   *****
   ```
6. Develop C programs for array insertion and display of array elements.
7. Write a C program to search for an element in a one-dimensional array.
8. Implement basic sorting on an array using bubble sort or selection sort.
9. Write a C program to delete an element from an array.
10. Develop a C program to display and concatenate two strings without using string library functions.
11. Write a C program to check whether a string is a palindrome.
12. Write C programs for string comparison and copying without using built-in functions.
13. Demonstrate simple pointer operations: accessing variable values using pointers.
14. Use pointers to perform addition of two numbers and pointer-based array traversal.
15. Develop a C program to dynamically allocate memory using malloc and initialize array values.
16. Write functions in C that accept arrays as arguments and return the sum of elements.
17. Write functions that take pointers as arguments and return pointer values.
18. Develop recursive functions to calculate factorial and Fibonacci numbers.
19. Write a C program to define a structure for a student containing the following fields: registration number, name, program name, and marks in three subjects. Read the details from the user and display them using structure variables.
20. Demonstrate the use of global and static variables within functions.

Suggested Readings:

B. S. Gottfried, Schaum's, "Outline of Programming with C," New York, NY, USA: McGraw-Hill, 3rd ed., 2017.

R. Thareja, "Computer Fundamentals and Programming in C," Oxford University Press, 3rd ed., 2023.

B. W. Kernighan and D. M. Ritchie, "The C Programming Language," Pearson Education, 2nd ed., 1988.

V. Rajaraman, "Computer Basics and C Programming," PHI Learning Pvt. Limited, 2006.

E. Balagurusamy, "Computer Concepts and Programming in C," McGraw Hill Education, 2nd ed., 2017.

J. R. Hanly and E. B. Koffman, "Problem Solving and Program Design in C," Pearson Education, 8th ed., 2015.

Y. Kanetkar, "Let Us C," BPB Publications, 21st ed., 2023.

H. M. Deitel and P. J. Deitel, "C: How to Program," Pearson Education, 8th ed., 2015.

S. Anami, N. Angadi, and S. Manvi, "Computer Concepts and Programming," PHI Learning Pvt. Ltd., 2nd ed., 2010.

R. S. Salaria, "Problem Solving and Programming in C," Khanna Publishing House 5th ed., 2016

Course Code : **BTC109** Course Type : **Audit Course** No. of Credits : **0** No. of Hours : **30**	**Course Title** **Sports and Yoga (Audit Course)**

Course Objectives:

- To make the students understand the importance of sound health and fitness principles as they relate to better health.
- To expose the students to a variety of physical and yogic activities aimed at stimulating their continued inquiry about Yoga, physical education, health and fitness.
- To develop among students an appreciation of physical activity as a lifetime pursuit and a means to better health**.**

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Demonstrate the Physical activities and Hatha Yoga focusing on yoga for strength, flexibility, and relaxation.
- **CO2**: To learn techniques for increasing concentration and decreasing anxiety which leads to stronger academic performance
- **CO3**: To understand and implement the basic skills associated with yoga and physical activities including strength and flexibility, balance and coordination.
- **CO4**: To develop understanding of health-related fitness components: cardiorespiratory endurance, flexibility and body composition etc.
- **CO5**: To improve personal fitness through participation in sports and yogic activities.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	–	–	–	–	–	–	–	–	2
CO2	3	3	2	–	–	–	–	–	–	–	–	2
CO3	3	3	2	–	–	–	–	–	–	–	–	2
CO4	2	3	2	–	–	–	–	–	1	–	–	3
CO5	3	3	3	2	2	–	–	–	1	–	–	3

Course Outline:

Course Contents:

Exercise 1. Introduction to Physical Education: Meaning & definition of Physical Education, Aims & Objectives of Physical Education, Changing trends in Physical Education.

Exercise II. Olympic Movement: Ancient & Modern Olympics (Summer & Winter), Olympic Symbols, Ideals, Objectives & Values, Awards and Honours in the field of Sports in India (Dronacharya Award, Arjuna Award, Dhayanchand Award, Rajiv Gandhi Khel Ratna Award etc.).

Exercise III. **Physical Fitness, Wellness & Lifestyle:** Meaning & Importance of Physical Fitness & Wellness, Components of Physical fitness, Components of Health-related fitness, Components of wellness, Preventing Health Threats through Lifestyle Change, Concept of Positive Lifestyle.

Exercise IV. Fundamentals of Anatomy & Physiology in Physical Education, Sports and Yoga: Define Anatomy, Physiology & Its Importance, Effect of exercise on the functioning of Various Body Systems. (Circulatory System, Respiratory System, Neuro-Muscular System etc.).

Exercise V. Yoga & Lifestyle: Asanas as preventive measures, Hypertension- Tadasana, Vajrasana, Pavan Muktasana, Ardha Chakrasana, Bhujangasana, Sharasana. Obesity: Procedure, Benefits & contraindications for Vajrasana, Hastasana, Ardh Matsyendrasana, Back Pain, Diabetes, Asthema Gomukhasana.

Suggested Readings:

Modern Trends and Physical Education by Prof. Ajmer Singh.
Light On Yoga by B.K.S. Iyengar.
Health and Physical Education – NCERT (11th and 12th Classes).

SEMESTER-II

Course Code : **BTC201** Course Type : **Basic Science** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Mathematics-II (Differential Equations and Vector Calculus)**

Course Objectives:

- To learn methods for solving first-order and higher-order ordinary differential equations and apply them to real-life problems.
- To understand the formation and solution of partial differential equations using standard techniques.
- To study and apply vector differential operators like gradient, divergence, and curl, and their physical interpretations, integral theorems such as Green's, Stokes', and Divergence theorems to solve engineering problems.

Course Outcomes (COs):

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

- **CO1**: Solve first-order and higher-order differential equations and apply them to real-world problems.
- **CO2**: Apply first-order differential equations to model real-world phenomena like cooling, growth, decay, and electrical circuits (RC/RL).
- **CO3**: Analyze and solve simultaneous linear differential equations and apply them to physical systems such as LCR circuits and simple harmonic motion.
- **CO4**: Formulate and solve partial differential equations using standard methods and compute line, surface, and volume integrals and interpret physical applications like work done, circulation, and flux
- **CO5**: Understand and apply concepts of vector differentiation and integration, including gradient, divergence, curl, and related theorems (Green's, Stokes', Divergence)

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	2	-	2	-	-	-	-	-	-	-	1
CO3	3	2	2	-	-	1	-	-	-	-	-	1
CO4	3	2	2	2	-	-	-	-	-	-	-	1
CO5	3	2	-	-	-	1	-	-	-	-	-	1

Course Outline:

Unit-I: **[10 Hours]**

First-Order Differential Equations and Applications: Linear differential equations, Bernoulli's equations, Exact differential equations, Equations reducible to exact form, Applications – Newton's law of cooling, Law of natural growth and decay, RC and RL circuits.

Unit-II: **[11 Hours]**

Higher-Order Linear Differential Equations and Applications: Linear differential equations with constant coefficients – Homogeneous and non-homogeneous forms, Complementary function, Particular integral, General solution, Wronskian, Method of variation of parameters, Simultaneous linear differential equations, Applications – LCR circuits, Simple harmonic motion.

Unit-III: **[08 Hours]**

Partial Differential Equations**:** Formation of partial differential equations by eliminating arbitrary constants and functions, First-order linear PDEs using Lagrange's method, Homogeneous linear PDEs with constant coefficients.

Unit-IV: **[08 Hours]**

Vector Calculus: Vector Differentiation – Scalar and vector point functions, Gradient, Directional derivative, Divergence, Curl, Vector identities.

Unit-V: **[08 Hours]**

Vector Integration – Line integral, Work done and circulation, Surface integral and flux, Volume integral, Theorems (without proofs) – Green's theorem, Stokes' theorem, Divergence theorem and simple applications.

Suggested Readings:

Higher Engineering Mathematics, B. S. Grewal, Khanna Publishers, 2017, 44^{th} Edition
Advanced Engineering Mathematics, Erwin Kreyszig, John Wiley & Sons, 2018, 10th Edition

References:

Thomas Calculus, George B. Thomas, Maurice D. Weir and Joel Hass, Pearson Publishers, 2018, 14^{th} Edition.
Advanced Engineering Mathematics, Dennis G. Zill and Warren S. Wright, Jones and Bartlett, 2018.
Advanced Modern Engineering Mathematics, Glyn James, Pearson publishers, 2018, 5^{th} Edition.
Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Alpha Science International Ltd., 2021 5^{th} Edition (9th reprint).
Higher Engineering Mathematics, B. V. Ramana, , McGraw Hill Education, 2017

Course Code : **BTC202** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Computer Aided Engineering Drawing**

Course Objectives:

- To introduce the fundamental principles of Engineering Drawing, enabling students to understand and create orthographic projections of various geometrical elements using both manual drafting techniques and computer-aided tools.
- To develop the ability to visualize and construct lateral surface developments of 3D objects, preparing students to apply these skills in solving practical design and manufacturing problems.
- To strengthen spatial visualization and drawing skills by training students to create accurate isometric drawings and convert them into corresponding orthographic views.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Produce orthographic projections of points and lines using both manual drafting methods and computer aided design tools.
- **CO2**: Create orthographic projections of planes and solids with accuracy using traditional and CAD techniques.
- **CO3**: Generate lateral surface developments of various solids for practical, real-world applications.
- **CO4.** Create isometric drawings and convert isometric views into precise orthographic projections.
- **CO5:** Create isometric views and projections of simple and combined solids using isometric scales and projection techniques.

COs /POs Mapping:

Cos/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	3	-	-	-	-	1	-	1
CO2	3	2	2	-	3	-	-	-	-	1	-	1
CO3	2	2	3	-	2	-	-	-	-	1	-	1
CO4	2	2	3	-	3	-	-	-	-	2	-	1
CO5	2	2	3	-	3	-	-	-	-	2	-	1

Course Outline:

Unit-I: **[10 Hours]**

Introduction: Significance of Engineering drawing, Free hand sketching of engineering drawing, Scales. Introduction to Computer Aided Drafting software, Co-ordinate system and reference planes HP, VP, RPP & LPP of 2D/3D environment. Selection of drawing sheet size and scale. Commands and creation of Lines, coordinate points, axes, square, rectangle, polygons, circles, ellipse, text, move, copy, off-set, mirror, rotate, trim, extend, chamfer, fillet and curves.

Orthographic Projections of Points, Lines: Introduction to Orthographic projections, Orthographic projections of points in 1st and 3rd quadrants. Orthographic projections of lines (Placed in all four quadrants)

Unit-II: **[11 Hours]**

Orthographic projections of planes: Triangular, square, rectangular, pentagonal, hexagonal and circular lamina (Placed in First quadrant only).

Unit-III: **[08 Hours]**

Orthographic Projection of Solids: Orthographic projection of right regular solids (Resting on HP only and inclined to both the planes); Prisms, Pyramids, Cones & Tetrahedraon.

Unit-IV: **[08 Hours]**

Development of Lateral Surfaces of Solids: Development of lateral surfaces of right regular Prisms, Pyramids, Cylinders & Cones and their frustums and truncations.

Unit-V: **[08 Hours]**

Isometric Views: Introduction to Isometric views, Isometric projections, Isometric scale. Isometric view of hexahedron (cube), right regular prisms, pyramids, cylinders, cones and spheres, Isometric view of combination of two simple solids

Suggested Readings:

N. D. Bhatt, V. M. Panchal, and P. R. Ingle, "Engineering Drawing," Charotar Publishing House, 55th ed.,2025.
L. P. Singh and H. Singh, "Engineering Drawing Principles and Applications," Cambridge University Press Education, 2021.

References:

B. Agrawal and C. M. Agrawal, “Engineering Graphics,” TMH Publication, 2012.
K. L. Narayana and P. Kannaiah, "Engineering Drawing," Scitech Publications (India) Pvt. Ltd., 2020.
CAD Software Theory and User Manuals (specific reference details needed). 2025.
K. Venugopal, *Engineering Drawing and Graphics + AutoCAD*, 4th ed., New Age International Publication Ltd., 2004.

Course Code : **BTC203** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Digital Logic Design**

Course Objectives:

- To understand the basic concepts of number systems, binary arithmetic, and digital code representations, principles of Boolean algebra and logic simplification techniques.
- To develop the ability to design and analyze combinational logic circuits.
- To apply the concepts of sequential logic and design of flip-flops, registers, and counters, programmable logic devices, and HDL fundamentals.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1**: Gain Knowledge on conversions between number systems and apply binary arithmetic and codes.
- **CO2**: Simplify and manipulate Boolean expressions using algebraic and graphical methods.
- **CO3**: Design and implement combinational circuits using standard logic modules.
- **CO4**: Analyze and design basic sequential circuits such as flip-flops, counters, and shift registers.
- **CO5**: Explain the architecture of memory and programmable logic devices, and describe simple HDL models.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	-
CO2	3	3	-	-	-	-	-	-	-	-	-	-
CO3	3	2	3	-	2	-	-	-	-	-	-	-
CO4	3	2	3	2	2	-	-	-	-	-	-	-
CO5	2	2	2	-	3	-	-	-	-	-	-	-

Course Outline:

UNIT-I **[09 hours]**

Number Systems, Codes, and Logic Fundamentals: Binary Numbers, Number-Base Conversions, Octal and Hexadecimal Numbers, Signed Binary Numbers – 1's and 2's Complement, Binary Codes – BCD, Gray Code, Excess-3, Binary Storage and Registers, Basic Definitions of Boolean Algebra, Basic Theorems and Properties, Boolean Functions, Canonical and Standard Forms, Other Logic Operations, Introduction to Hardware Description Languages (HDLs) and their syntax.

UNIT-II **[10 hours]**

Boolean Expression Simplification and Minimization**:** Simplification of Boolean expressions – Algebraic methods and Canonical Forms, Gate-level Minimization, K-Map method – Three-variable map, Four-variable map, Five-variable map, Product of Sums simplification, Don't-care conditions, NAND and NOR implementation, Quine-McCluskey algorithm, Determination, and selection of Prime Implicants, Essential and Non-essential Prime Implicants.

UNIT-III **[09 hours]**

Combinational Logic Circuits: Combinational Logic, Implementing Combinational Logic, Binary Adder-Subtractor, Decimal Adder, Binary Multiplier, Parallel Adders and Look-ahead Adders, Comparator, Decoders, Encoders, Multiplexers, Demultiplexers, Parity Generators and Checkers.

UNIT-IV **[09 hours]**

Sequential Circuits and Memory Devices: Latches and Flip-Flops – SR, JK, D, T, Excitation Tables, Shift Registers, Design of Counters – Ripple Counters, Synchronous Counters, Memory Units, Integrated Circuits – MOS, CMOS, RTL, DTL, TTL, ECL

UNIT-V **[08 hours]**

Programmable Logic Devices – PLA, PAL, Designing Combinational Logic Circuits with PLDs, Random Access Memory (RAM), Read-Only Memory (ROM), Content Addressable Memory (CAM).

Suggested Readings:

M. M. Mano, "Digital Logic and Computer Design," Pearson Education India, 2023.

T. L. Floyd, "Digital Fundamentals," Pearson Education India, 11th ed., 2023.

References:

J. F. Wakerly, "Digital Design: Principles and Practices," Pearson Education India, 5th ed., 2018.

C. R. Kime and M. M. Mano, "Logic and Computer Design Fundamentals," Pearson Education India, 5th ed., 2015.

N. S. Widmer and R. J. Tocci, "Digital Systems: Principles and Applications," Pearson Education India, 10th ed., 2017

Course Code : **BTC204** Course Type : **Core Course** No. of Credits **: 4.00** No. of Hours : **45 + 30 Lab**	**Course Title** **Python Programming**

Course Objectives:

- To understand the basic concepts on python data types, python programming.
- To acquire skills for developing python programming and algorithms for data handling.
- To learn the python concepts, principles, functions and develop an application.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** To analyze the basic concepts and working principles of Python Programming.
- **CO2:** To develop algorithmic solutions to simple computational problems.
- **CO3:** To understand the structure of solving problems using programming.
- **CO4:** To acquire skills the concepts of compound data using Python lists, tuples, dictionaries.
- **CO5:** To explore the various multimedia features using python.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	–	–	2
CO2	3	3	2	–	–	–	–	–	–	–	–	3
CO3	3	2	2	–	–	–	–	–	–	–	–	3
CO4	3	3	2	–	–	–	–	–	–	–	–	3
CO5	3	2	3	2	2	–	–	–	–	–	–	3

Course Outline:

UNIT-I **[10 hours]**

Introduction: History - Features - Working with Python - Installing Python - basic syntax - Data types - variables - Manipulating Numbers - Text Manipulations - Python Built-in Functions.

UNIT-II **[15 hours]**

components of python programming: Python objects and other languages - Operator Basics Numbers - String - List - Tuples - Dictionaries - Files - Object Storage - Type Conversion - Type Comparison – Statements, Assignments - Control Statements.

UNIT-III **[15 hours]**

functions and modules: Functions Definition and Execution - Arguments - Return Values - Advanced Function Calling - Modules - Importing modules - Packages - Creating a module.

UNIT-IV **[20 hours]**

Object oriented and exception handling: Classes and Objects - creating a class - class methods -class inheritance. Exceptions Handling-Build in Exceptions- Files, File operations, reading a file content, writing a file, change position, controlling file I/O, Manipulating file paths.

UNIT-V **[15 hours]**

Applications: Working with PDF and Word Documents - Working with CSV Files and JSON Data - Sending Email and Text Messages - Manipulating Images - Using Python for Multimedia.

Suggested Readings:

A. B. Downey, “Think Python: How to Think Like a Computer Scientist,” Shroff O'Reilly Publishers, 3rd ed., 2019.

G. Van Rossum and F. L. Drake Jr., “The Python Language Reference Manual,” Network Theory Ltd., 2023.

References:

M. C. Brown, “The Complete Reference – Python,” Tata McGraw Hill Indian Edition, 2018.

E. Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction to Programming,” 2nd ed., No Starch Press, 2019.

T. A. Budd, “Exploring Python,” Tata McGraw Hill Education, 2nd ed., 2016.

R. Sedgewick, K. Wayne, and R. Dondero, “Introduction to Programming in Python: An Interdisciplinary Approach,” Pearson India Education Services Pvt. Ltd., 2nd ed., 2023.

Course Code : **BTC205** Course Type : **Humanities** No. of Credits **: 2.00** No. of Hours : **30**	**Course Title** **Design Thinking**

Course Objectives:

- To understand the basic concepts on engineering as a tool for social and economic development.
- To acquire skills for developing introduce the design thinking process through real-world contexts.
- To foster creativity and innovation through ideation and prototyping, testing and feedback loops.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the importance of engineering in society.
- **CO2:** Apply the design thinking process to real-world problems.
- **CO3:** Define clear problem statements based on user needs.
- **CO4:** Use creativity tools to generate multiple ideas.
- **CO5:** Create and improve prototypes based on feedback

COs /POs Mapping:

Cos/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	-	2	-	-	-	2	-	-
CO2	3	2	2	-	-	2	-	-	-	-	-	-
CO3	3	2	3	1	2	-	-	-	-	-	-	-
CO4	3	2	3	-	2	-	-	-	-	-	-	-
CO5	3	2	3	2	2	-	-	-	1	-	-	-

Course Outline:

UNIT-I **[06 hours]**

Introduction to Engineering**:** "Engineering" as a vehicle for social and economic development; the impact of science/engineering on our day-to-day lives; the process of engineering a product; various career options.

UNIT-II **[06 hours]**

Introduction and identifying the need: Understanding the unique needs of the user - empathize - define - ideate - prototype - test. Problem Formulation: Framing a problem statement neutrally using adequate checks. Case studies.

UNIT-III **[06 hours]**

Case Studies - Develop an appreciation for the design process and its application in specific settings (Guest lectures, Videos, Field visits, Interplay lectures of design-based movies).

UNIT-IV **[06 hours]**

Concept Generation: Generate multiple concepts using various creativity tools and thinking styles, Prototyping: Select from ideas and make quick prototypes (mock-ups) using available material.

UNIT-V **[06 hours]**

Evaluation: Iterative process of ideation, prototyping and testing-Take the mock-ups to users for feedback and iterate the process till users feel delighted.

Suggested Readings:

Pressman, "Design Thinking: A Guide to Creative Problem Solving for Everyone," Routledge Taylor and Francis Group, 1st ed., 2019.

T. Brown, "Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation," Harper Business, 2009.

References:

G. E. Dieter and L. C. Schmidt, "Engineering Design," McGraw-Hill Education, 5th ed., 2019

K. Ulrich, S. Eppinger, and M. Yang, "Product Design and Development," McGraw-Hill Education, 7th ed., 2020.

Course Code : **BTC206** Course Type : **Humanities** No. of Credits : **2.00** No. of Hours : **30**	**Course Title** **Universal Human Values**

Course Objectives:

- To help students understand the need and importance of value-based living.
- To facilitate self-exploration and develop a holistic perspective of life and profession.
- To promote awareness of value-based education, holistic technologies, sustainable management practices, and professional ethics for building a humane and harmonious society.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand the concepts of right understanding, human relationships, and physical needs.
- **CO2:** Analyze the harmony within themselves and with the body for healthy and ethical living.
- **CO3:** Apply values to develop harmony in family, society, and nature.
- **CO4:** Evaluate the importance of universal human values and sustainable living.
- **CO5:** Demonstrate responsible and ethical behaviour in personal and professional contexts.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	-	-	3	2	2	1	-	-	-
CO2	2	3	-	-	-	3	1	2	-	-	-	-
CO3	1	3	-	-	-	3	3	2	2	-	-	-
CO4	1	3	-	-	-	3	3	1	2	-	-	-
CO5	3	2	3	1	2	3	3	3	3	2	2	2

Course Outline:

UNIT-I **[06 hours]**

Introduction to Value Education and Human Aspirations: Understanding Value Education, Self-exploration as the Process for Value Education, Right Understanding - Relationship - Physical Facility (Holistic Development and the Role of Education), Continuous Happiness and Prosperity as Basic Human Aspirations.

UNIT-II **[06 hours]**

Harmony in Human Being, Family and Society: Understanding Human Being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Harmony in the Family - the Basic Unit of Human Interaction, Trust - the Foundational Value in Relationships, Respect - Right Evaluation .

UNIT-III **[06 hours]**

Understanding Harmony in Society, Vision for the Universal Human Order, Harmony in Nature and Implications of Holistic Understanding**:** Understanding Harmony in Nature, Interconnectedness - Self-regulation - Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, Holistic Perception of Harmony in Existence -Describing - Defining – Classifying.

UNIT-IV **[06 hours]**

Writing Introduction and Conclusion, Natural Acceptance of Human Values, Definitiveness of Ethical Human Conduct, Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics

UNIT-V **[06 hours]**

Holistic Technologies - Production Systems - Management Models (Case Studies), Strategies for Transition toward a Value-based Life and Profession.

Suggested Readings:

P. Kapoor, "Professional Ethics and Human Values," Khanna Book Publishing Company, New Delhi, 2022.

R. R. Gaur, R. Asthana, and G. P. Bagaria, "The Textbook - A Foundation Course in Human Values and Professional Ethics," Excel Books, New Delhi, 2nd revised ed., 2019.

References:

R. R. Gaur, R. Asthana, and G. P. Bagaria, "The Teacher's Manual - Teachers Manual for A Foundation Course in Human Values and Professional Ethics," 2nd revised ed., Excel Books, New Delhi, 2019.

A. Leonard, "The Story of Stuff," 2011.

A. N. Tripathi, "Human Values," New Age International Publishers, New Delhi, 2019.

M. K. Gandhi, "The Story of My Experiments with Truth," FP Classic, 2009.

A. Nagaraj, "VanVidya: Ek Parichaya," "Jeevan Vidya Prakashan," Amarkantak, 1999

Course Code : **BTC207** Course Type : **Core Lab** No. of Credits **: 2.00** No. of Hours : **60**	**Course Title** **Digital Logic Design Lab**

Course Objectives:

- Understand the different types of adder circuits, logic gates, and Boolean algebra.
- Learn to performs the opposite function, signal and routing.
- Design, Develop & analyze combinational logic circuits, operation of flip-flops and sequential circuits, test digital circuits using simulation tools.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Implement number system conversions and basic logic circuits.
- **CO2:** Simplify Boolean expressions and design optimized logic circuits.
- **CO3:** Design and simulate combinational logic components like adders, encoders, decoders, and multiplexers.
- **CO4:** Analyze and construct sequential logic systems including counters and shift registers.
- **CO5:** Use digital circuit simulation software to test and verify logic designs.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	2	2	-	-	-	1	-	-	-
CO2	3	2	3	3	2	-	-	-	1	-	-	-
CO3	3	2	3	3	2	-	-	-	2	-	-	1
CO4	3	3	3	2	2	-	-	-	2	-	-	1
CO5	2	2	3	3	3	-	-	-	1	1	-	1

List of Experiments: (Through Virtual Labs)

1. To understand and implement different types of adder circuits used in digital systems.
2. A demultiplexer (DEMUX) performs the opposite function, taking a single input signal and routing it to one of many output lines.
3. To understand and implement BCD to 7-segment decoder circuits used in digital display systems.
4. To understand and implement an Arithmetic Logic Unit (ALU) - a fundamental component that performs arithmetic and logical operations in digital systems.
5. To understand and implement digital comparator circuits used in digital systems for magnitude comparison operations.
6. To understand and implement different types of latches and flip-flops used in sequential digital circuits.
7. To understand and implement different types of register circuits used in digital systems.
8. To understand and implement different types of counter circuits used in digital systems.
9. To understand and implement different types of multiplier circuits used in digital systems.
10. To understand and implement finite state machines (FSMs) and their state diagrams in digital systems.

Suggested Readings:

M. M. Mano, "Digital Logic and Computer Design," Pearson Education India, 2023.

T. L. Floyd, "Digital Fundamentals," Pearson Education India, 11th ed., 2023.

J. F. Wakerly, "Digital Design: Principles and Practices," Pearson Education India, 5th ed., 2018.

C. R. Kime and M. M. Mano, "Logic and Computer Design Fundamentals," Pearson Education India, 5th ed., 2015.

N. S. Widmer and R. J. Tocci, "Digital Systems: Principles and Applications," Pearson Education India, 10th ed., 2017.

C. H. Roth Jr., L. L. Kinney, and E. B. John, "Fundamentals of Logic Design," Cengage Learning, 8th ed., 2020.

Course Code : **BTC208** Course Type : **Core Lab** No. of Credits : **1.00** No. of Hours : **30**	**Course Title** **IT Workshop**

Course Objectives:

- To Study the components of computer, Operating systems, software's, document design tools.
- To provide skills in design of documentations, and Productivity tools including Word, Excel, PowerPoint and Publisher.
- To gain knowledge on modules tasks includes training on PC Hardware, Internet & World Wide Web

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Identify and use basic Hardware troubleshooting
- **CO2:** Gain knowledge on Understand Hardware components and inter dependencies
- **CO3:** Apply skills to Safeguard computer systems from viruses/worms.
- **CO4:** Implement Document/ Presentation preparation.
- **CO5:** Evaluate the Perform of calculations using spreadsheets.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	1	1	-	-	2	-	-	-
CO2	2	3	3	2	2	1	-	-	2	-	-	-
CO3	2	2	3	2	2	1	-	-	1	-	-	1
CO4	2	2	3	2	3	1	1	-	1	-	-	1
CO5	2	1	2	1	2	3	1	1	2	2	1	1

List of Task Exercises

Task 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Task 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Task 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Task 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both Windows and Linux. Lab instructors should verify the installation and follow it up with a Viva.

Task 5: Students will be working on basic power point utilities and tools which help them create basic PowerPoint presentations. PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in PowerPoint.

Task 6: Interactive presentations - Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts.

Task 7: Creating a Scheduler - Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text.

Task 8: Calculating GPA -. Features to be covered: - Cell Referencing, Formulae in excel – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, LOOKUP/VLOOKUP.

Task 9: Internet & World Wide Web, Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting.

Task 10: Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides.

Task 11: Creating project abstract Features to be covered: -Formatting Styles, inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes. (Student should submit a project report to the instructor).

Suggested Readings:

Comdex Information Technology course tool kit Vikas Gupta, WILEY Dreamtech.

The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, WILEY Dreamtech.

Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.

References:

PC Hardware - A Handbook – Kate J. Chase PHI (Microsoft).

LaTeX Companion – Leslie Lamport, PHI/Pearson.

IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.

Course Code : **BTC209** Course Type : **Audit Course** No. of Credits **: 0** No. of Hours : **30**	**Course Title** **National Service Scheme (NSS)**

Course Objectives:
National Service Scheme (NSS) will enable the students to:
- Understand the community in general in which they work.
- Identify the needs and problems of the community and involve them in problem –solving.
- Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems.
- Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.
- Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony in general.

Course Outcomes:
At the end of the course, the student will be able to:
- **CO1:** Understand the importance of his / her responsibilities towards society.
- **CO2:** Analyse the environmental and societal problems/issues and will be able to design solutions for the same
- **CO3:** Evaluate the existing system and to propose practical solutions for the same for sustainable development
- **CO4:** Implement government or self-driven projects effectively in the field.
- **CO5:** Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony in general.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	3	2	1	1	-	-	-
CO2	2	3	3	2	-	1	2	-	3	2	1	2
CO3	0	0	2	2	-	2	1	2	2	1	2	2
CO4	1	1	1	1	1	1	1	-	1	-	-	1
CO5	2	-	2	-	-	2	2	1	3	3	2	2

Course Outline:

1. Organic farming, Indian Agriculture (Past, Present and Future) Connectivity for marketing.
2. Waste management– Public, Private and Govt organization.
3. Setting of the information imparting club for women leading to contribution in social and economic issues.
4. Water conservation techniques – Role of different stakeholders– Implementation.
5. Preparing an actionable business proposal for enhancing the village income and approach for implementation.
6. Helping local schools to achieve good results and enhance their enrolment in Higher/ technical/ vocational education.
7. Developing Sustainable Water management system for rural areas and implementation approaches.
8. Contribution to any national level initiative of Government of India. For eg. Digital India, Skill India, Swatch Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc.
9. Spreading public awareness under rural outreach programs. (minimum 5 programs).
10. Social connect and responsibilities.
11. Plantation and adoption of plants. Know your plants.
12. Organize National integration and social harmony events /workshops /seminars. (Minimum 02 programs).
13. Govt. school Rejuvenation and helping them to achieve good infrastructure.

NOTE:
1. Student/s in individual or in a group Should select any one activity in the beginning of each semester till end of that respective semester for successful completion as per the instructions of NSS officer with the consent of HOD of the department.
2. At the end of every semester, activity report should be submitted to the department.

Suggested Readings:

NSS Course Manual, Published by NSS Cell Govt Of India, Ministry of youth affairs & sports, New Delhi.
Government of India, NSS cell, Activities reports and its manual.
Government of Andhra Pradesh, NSS cell, activities reports and its manual.

SEMESTER-III

Course Code : **BTC301** Course Type : **Basic Science** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Mathematics-III (Probability & Statistics)**

Course Objectives:

- To introduce the fundamental concepts of probability, random variables, and distributions.
- To develop logical reasoning skills for modeling uncertainty and variability in data using probability and random variables.
- To equip students with statistical tools for hypothesis testing, correlation, regression, and real-world data analysis**.**

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand and apply basic probability principles, conditional probability, and Bayes' theorem.
- **CO2:** Analyze discrete and continuous random variables with their distributions and joint behaviors.
- **CO3:** Compute expectations, variances, and apply central limit theorem and related inequalities.
- **CO4:** Conduct hypothesis testing for large and small samples using standard statistical tests.
- **CO5:** Interpret relationships between variables using correlation and regression techniques.

COs /POs Mapping:

Cos/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	1	-	-	-	1	-	-	-
CO2	3	3	3	2	1	-	-	-	1	-	-	1
CO3	3	2	2	2	1	-	-	-	1	-	-	1
CO4	3	3	3	3	2	-	-	-	2	-	-	1
CO5	3	2	3	2	2	-	-	-	2	-	-	1

Course Outline:

UNIT-I **[12 hours]**

Introduction to Probability, Sets, Events, Axioms of Probability, Conditional Probability and Independence, Bayes Theorem.Random Variables, Definitions, Cumulative Distribution Functions, Probability Mass Function, Probability Density Function, Joint and Conditional Distributions.

UNIT-II **[12 hours]**

Expectations and Distributions: Expectations, Mean, Variance, Moments, Correlation, Chebychev and Schwarz Inequalities, Moment-Generating and Characteristic Functions, Chernoff Bounds, Conditional Expectations, Law of Large Numbers, Central Limit Theorem, Uniform Distribution, Binomial Distribution, Poisson Distribution, Normal Distribution.

UNIT-III **[06 hours]**

Test for Large Samples: Testing of Hypothesis, Null and Alternate Hypothesis, Level of Significance and Critical Region, Z-test for Single Mean and Difference of Means, Z-test for Single Proportion and Difference of Proportions.

UNIT-IV **[06 hours]**

Test for Small Samples: t-test for Single Mean and Difference of Means, F-test for Comparison of Variances, Chi-square Test for Goodness of Fit, Chi-square Test for Independence.

UNIT-V **[09 hours]**

Correlation, Lines of Regression, Examples and Applications of Regression in Data Analysis.

Suggested Readings:

S. C. Gupta and V. K. Kapoor, "Fundamentals of Mathematical Statistics", 12^{th} Edition, S. Chand & Co, 020.
R. A. Johnson, Miller, and Freund's "Probability and Statistics for Engineers", Pearson Publishers, 9^{th} Edition, 2017

References:

John E. Freund Emeritus, Gary A. Simon, Modern Elementary Statistics, PHI, 9^{th} Edition, 2006,
V.K. Rohatgi and A.K. Md. Ehsanes Saleh, An Introduction to Probability theory and Mathematical Sciences, Wiley, 2001.
Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Academic Press, 5^{th} Edition 2014

Course Code : **BTC302** Course Type : **Core Course** No. of Credits **: 4.00** No. of Hours : **45T+ 30L**	**Course Title** **Object Oriented Programming through JAVA**

Course Objectives:

- To provide a solid foundation in object-oriented programming concepts using Java.
- To develop the ability to design modular Java programs using classes, inheritance, interfaces, and packages.
- To enable students to build reliable Java applications using exception handling, multithreading, file I/O, and string processing techniques.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand and apply object-oriented programming principles using Java syntax and structures.
- **CO2:** Implement inheritance, polymorphism, interfaces, and packages to build modular applications.
- **CO3:** Handle exceptions and develop multithreaded programs for responsive and concurrent execution.
- **CO4:** Use generics and file I/O to create flexible and data-driven applications.
- **CO5:** Develop Java programs that use multithreading, file handling, and string manipulation for efficient and concurrent execution.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	2	-	-	-	1	-	-	1
CO2	3	3	3	2	2	-	-	-	1	-	-	1
CO3	3	3	3	3	2	-	-	-	2	-	-	1
CO4	3	2	3	3	2	-	-	-	1	-	-	1
CO5	3	2	3	3	3	-	-	-	2	-	-	1

Course Outline:

UNIT-I **[20 hours]**

Fundamentals of OOP and Java Programming**:** Overview of OOP, Object Oriented Programming Paradigms, Features of Object-Oriented Programming, Java Buzzwords, Overview of Java, Data Types, Variables and Arrays, Operators, Control Statements, Programming Structures in Java, Defining Classes in Java, Constructors, Methods, Access Specifiers.

UNIT-II **[15 hours]**

Classes and Inheritance: Overloading Methods, Objects as Parameters, Returning Objects, Static, Nested and Inner Classes, Inheritance Basics, Types of Inheritance, Super Keyword, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Final with Inheritance.

UNIT-III **[10 hours]**

Packages, and Interfaces: Packages, Packages and Member Access, Importing Packages, Interfaces, default interface methods, use of static methods in interfaces.

UNIT-IV **[15 hours]**

Exception Handling, Multithreading and String Handling: Exception Handling Basics, Multiple Catch Clauses, Nested try Statements, Java's Built-in Exceptions, User-defined Exceptions.

UNIT-V **[15 hours]**

Java Thread Model, creating a Thread, Multiple Threads, Thread Priorities, Synchronization, Inter-thread Communication, I/O Basics, Reading and Writing Console I/O, Reading and Writing Files, Strings, Basic String Class Methods, StringBuffer Class.

Suggested Readings:

Herbert Schildt, "Java: The Complete Reference", McGraw Hill Education, 11*th* Edition, 2019.
Cay S. Horstmann, "Core Java Fundamentals", Volume 1, Prentice Hall, 11*th* Edition, 2018.

References:

Timothy Budd, "Object Oriented Programming with JAVA", Updated Edition, Pearson Education, 2009

Course Code : **BTC303** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Data Structures**

Course Objectives:

- To introduce fundamental data structures such as stacks, queues, linked lists, trees, heaps, and graphs, along with their implementation techniques.
- To develop algorithmic thinking and problem-solving skills using iterative and recursive strategies.
- To equip students with the ability to analyze and apply appropriate data structures and algorithms for efficient computation and software development

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand and implement basic linear data structures like stacks, queues, and linked lists.
- **CO2:** Apply tree-based structures and balanced search trees to solve hierarchical data problems.
- **CO3:** Utilize hashing, heaps, and disjoint sets for efficient data access and manipulation.
- **CO4:** Implement and analyze various sorting techniques and their computational efficiency.
- **CO5:** Apply graph traversal and shortest path algorithms to solve real-world problems.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	2	-	-	-	1	-	-	1
CO2	3	3	3	3	2	-	-	-	1	-	-	1
CO3	3	3	3	3	3	-	-	-	2	-	-	1
CO4	3	3	3	3	3	-	-	-	1	-	-	1
CO5	3	3	3	3	3	-	-	-	2	-	-	1

Course Outline:

UNIT-I **[09 hours]**

Introduction to Abstract Data Types (ADTs), Implementation and Applications of Stacks, implementation, Operations and Applications of Queues, Array Implementation of Circular Queues, Linked Lists, Search and Update Operations on Varieties of Linked Lists, Linked List Implementation of Stacks and Queues.

UNIT-II **[10 hours]**

Introduction to Trees, Implementation of Trees, Binary Trees, Tree Traversals with an Application, Binary Search Trees (BSTs), Query and Update Operations on BSTs, AVL Trees, Rotations, Search and Update Operations on Balanced BSTs, Splay Trees, B-trees.

UNIT-III **[08 hours]**

Hashing, Implementation of Dictionaries, Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing, Analysis of Search Operations, Priority Queues, Priority Queue ADT, Binary Heap Implementation and Applications of Priority Queues, Disjoint Sets.

UNIT-IV **[08 hours]**

Sorting Algorithms, Insertion Sort, Merge Sort, Quick Sort, Heap Sort.

UNIT-V **[10 hours]**

Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's Algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths.

Suggested Readings:

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Second Edition, PHI, 2009.
Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition, Pearson Education, 2006.

References:

Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2011.
Michael T.Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples", Second Edition, Wiley-India, 2006.

Course Code : **BTC304** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Computer Organization and Architecture**

Course Objectives:

- To introduce the basic principles and building blocks of computer organization and the design of instruction set architectures.
- To develop an understanding of processor design, pipelining, and instruction-level parallelism.
- To explore the memory hierarchy, caching strategies, and input/output systems with performance considerations.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Describe the fundamental functional units of a computer and their roles in execution.
- **CO2:** Interpret and analyze various instruction set architectures including RISC-V and x86.
- **CO3:** Design and evaluate single-cycle and multi-cycle processor architectures with pipelining and hazard resolution.
- **CO4:** Analyze memory hierarchy concepts including caching, locality of reference, and performance optimization.
- **CO5:** Explain I/O mechanisms such as memory-mapped I/O, interrupts, and DMA for data transfer.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	-	-	-	-	1	-	-	1
CO2	3	3	2	2	-	-	-	-	1	-	-	1
CO3	3	3	3	3	2	-	-	-	1	-	-	1
CO4	3	3	3	3	3	-	-	-	1	-	-	1
CO5	3	3	2	2	2	-	-	-	1	-	-	1

Course Outline:

UNIT-I **[08 hours]**

Role of abstraction, Basic functional units of a computer, Von-Neumann model of computation, Moore's law, Notion of performance, Data representation, and basic operations.

UNIT-II **[08 hours]**

Instruction Set Architecture: CPU registers, Instruction format and encoding, addressing modes, Instruction set, Instruction types, Instruction decoding and execution, Basic instruction cycle.

UNIT-III **[09 hours]**

Processor Design and Architecture: RISC and CISC concepts, RISC-V instructions, X86 instruction set. Clocking methodology, Amdahl's law, Building a data path and control, Single cycle processor, multi-cycle processors.

UNIT-IV **[09 hours]**

Pipelining and Instruction Level Parallelism: Instruction pipelining, Instruction Level Parallelism (ILP), Data hazards, Control hazards, Hazard mitigations, Limits of ILP.

UNIT-V **[10 hours]**

Memory and I/O Systems: SRAM and DRAM, Locality of reference, caching techniques, indexing mechanisms, Cache block size and associativity, Cache size trade-offs, Write-through and write-back caches, Cache replacement policies, Memory interleaving, Magnetic disks, Flash memory, I/O mapped I/O, Memory-mapped I/O, Programmed I/O, Interrupt-driven I/O, Direct Memory Access (DMA).

Suggested Readings:

Carl Hamacher, "Computer Organization and Embedded Systems", McGrawHill Higher Education, 6*th* Edition, 2022.

David A. Patterson and John L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface", Elsevier, 5*th* Edition, 2014.

References:

Vincent P. Heuring and Harry F. Jordan, "Computer System Design and Architecture", Pearson Education, 2nd Edition, 2016.

Smruti Ranjan Sarangi, "Computer Organisation & Architecture", McGraw Hill, 2014.

Mano M. Morris, "Computer System Architecture", Pearson, 2007

Course Code : **BTC305** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Discrete Mathematics**

Course Objectives:

- To understand the fundamental concepts of logic, proof techniques, and propositional and predicate logic.
- To apply combinatorial techniques and recurrence relations for solving counting problems and analyzing algorithms.
- To analyze discrete structures such as graphs and trees, including graph coloring and domination concepts, and their applications.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Apply logical reasoning and methods of proof to validate mathematical statements.
- **CO2:** Analyze and solve problems involving various types of graphs and trees.
- **CO3:** Solve recurrence relations using standard methods and apply them to computer science problems.
- **CO4:** Use combinatorial techniques for solving enumeration and counting problems.
- **CO5:** Evaluate graph properties such as chromatic number and domination, and their applications.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	2	-	-
CO2	3	3	2	-	2	-	-	-	-	1	-	-
CO3	3	3	2	-	1	-	-	-	-	2	-	-
CO4	3	2	-	-	-	-	-	-	-	2	-	-
CO5	3	2	2	-	2	-	-	-	-	1	-	-

Course Outline:

UNIT-I **[09 hours]**

Mathematical Logic and Normal Forms: Statements and Notation, Connectives, Functionally Complete Set of Connectives, Methods of Proof of an Implication, Inference Theory of Propositional Logic, Normal Forms, Limitations of Propositional Logic. Quantified Propositions, Predicate Logic and other Methods of Proof, Rules of Inference for Quantified Propositions.

UNIT-II **[12 hours]**

Graph Theory: Basic Definitions, Representation of Graphs, Connected Components, Connectivity of Graphs, Block Graphs, Trees, Graph Isomorphism, Planar Graphs, Euler's Formula, Eulerian Graphs, Hamiltonian Graphs, Chromatic Number of a Graph, Brook`s Theorem, Planar Graph Coloring, Applications of Graph Coloring, Domination Number of a Graph, Bounds on Domination Number, Applications of Domination.

UNIT-III **[07 hours]**

Recurrence Relations: Solving Recurrence Relations by Substitution Method, The Method of Characteristic Roots, Solutions of Homogeneous and Inhomogeneous Recurrence Relations using Characteristic Roots Method, Closed Form Formula for nth Fibonacci Number, Counting Minimum Number of Nodes in an AVL Tree of Given Height.

UNIT-IV **[07 hours]**

Generating Functions of Sequences, Solving Recurrence Relations using Generating Functions, Counting Number of Binary Search Trees using Generating Functions**.**

UNIT-V **[10 hours]**

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion-Exclusion, Stirling Numbers of the Second Kind.

Suggested Readings:

Kenneth H. Rosen, "Discrete Mathematics and Its Applications with Combinatorics and Graph Theory", McGraw Hill Education, 2011, 7th Edition.

References:

Joe L. Mott, Abraham Kandel, Theodore P. Baker, "Discrete Mathematics for Computer Scientists and Mathematicians", PHI, 2001, 2nd Edition.

Gary Chartrand and Ping Zhang, Introduction to Graph Theory, McGraw-Hill Higher Education, 2006, 1st Edition.

Tremblay J. P. and Manohar R., "Discrete Mathematical Structures", McGraw Hill Education, 2017, 1st Edition.

Course Code : **BTC306** Course Type : **Skill Course** No. of Credits **: 2.00** No. of Hours : **45**	**Course Title** **Introduction to R Programming**

Course Objectives:

- To introduce R as a programming language for data analysis and statistical computing.
- To enable students to perform data manipulation, cleaning, and visualization using R.
- To build foundational skills in applying basic statistical techniques using R.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand and use basic syntax, data structures, and control statements in R.
- **CO2:** Import, clean, and manipulate data using R libraries such as dplyr and tidyr.
- **CO3:** Create data visualizations using ggplot2 and base plotting functions.
- **CO4:** Perform basic statistical analysis including correlation, hypothesis testing, and regression.
- **CO5:** Write simple R programs to solve real-world data problems.

COs /POs Mapping:

Cos/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	-	-	-	-	1	-	1
CO2	3	3	2	2	2	-	-	-	1	1	-	1
CO3	3	3	2	2	2	-	-	-	1	1	-	1
CO4	3	3	3	3	2	-	-	-	1	1	-	1
CO5	3	2	3	3	2	-	-	-	2	1	-	1

Course Outline:

UNIT-I **[10 hours]**

Basics of R Programming and Data Structures**:** R environment and RStudio, Variables and data types, Vectors, Lists, Matrices, Data frames, Factors, Reading and writing files, Basic input/output, Using built-in functions and basic data handling.

UNIT-II **[09 hours]**

Data Manipulation: Data wrangling with dplyr, Data tidying with tidyr, Filtering and transforming datasets, Handling missing values.

UNIT-III **[08 hours]**

Data Visualization: Visualizing data using base R and ggplot2, Bar charts, Histograms, Box plots, Scatter plots, Plot customization

UNIT-IV **[08 hours]**

Exploratory Data Analysis and Basic Statistics: Descriptive statistics (mean, median, mode, variance, standard deviation), Correlation analysis, Hypothesis testing (t-test, chi-square), Introduction to regression simple linear regression.

UNIT-V **[10 hours]**

Control structures (if, loops), Writing functions, Mini project, or case study on real dataset

Suggested Readings:

H. Wickham and G. Grolemund, *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2016.

References:

N. Matloff, *The Art of R Programming: A Tour of Statistical Software Design*, 1st ed. San Francisco, CA, USA: No Starch Press, 2011.

G. Grolemund, *Hands-On Programming with R: Write Your Own Functions and Simulations*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2014.

A. Field, J. Miles, and Z. Field, *Discovering Statistics Using R*, 1st ed. London, U.K.: SAGE Publications, 2012

Course Code : **BTC307** Course Type : **Core lab** No. of Credits **: 2.00** No. of Hours : 60	**Course Title** **Data Structures Lab**

Course Objectives:

- To practice object-oriented programming concepts such as constructors, inheritance, interfaces, and exception handling through programs.
- To implement fundamental data structures like arrays, stacks, queues, linked lists, and trees using programs.
- To understand and implement searching and sorting algorithms through hands-on programming.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Write programs using object-oriented programming concepts such as constructors, inheritance, interfaces, and abstract classes.
- **CO2:** Develop programs using arrays, type conversions, and input/output streams.
- **CO3:** Apply exception handling techniques to create reliable and error-free programs.
- **CO4:** Implement linear and non-linear data structures including stacks, queues, circular queues, linked lists, and binary search trees.
- **CO5:** Implement and analyze searching and sorting algorithms such as binary search, bubble sort, insertion sort, selection sort, quick sort, and merge sort.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	2	2	-	-	-	1	1	-	2
CO2	3	2	3	3	3	-	-	-	2	2	-	2
CO3	3	2	3	2	3	-	-	-	1	1	-	2
CO4	3	3	3	3	2	-	-	-	1	1	-	2
CO5	3	3	3	3	3	-	-	-	1	1	-	2

List of Experiments:

1. Program to implement constructors and destructors with array of objects.
2. Program to demonstrate function overloading.
3. Program to implement different types of inheritances like multiple, Multilevel and hybrid.
4. I/O Program to demonstrate the use of abstract classes.
5. Program to demonstrate I/O streams and functions.
6. Program to perform all possible type conversions.
7. Program to demonstrate exception handling technique.
8. Program to implement Arrays.
9. Program to implement interfaces.
10. Program to implement binary search
11. Program to implement Stack.
12. Program to implement Queues.
13. Program to implement circular queue.
14. Program to implement linked list: singly, doubly and circular.
15. Program to implement Binary Search Tree using array.
16. Program to implement bubble sort.
17. Program to implement insertion sort
18. Program to implement selection sort
19. Program to implement quick sort
20. Program to implement merge sort

Suggested Readings:

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Second Edition, PHI, 2009.
Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition, Pearson Education, 2006.

References:

Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2011.
Michael T.Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples", Second Edition, Wiley-India, 2006

Course Code : **BTC308** Course Type :**Core Course Lab** No. of Credits **: 2.00** No. of Hours : **60**	**Course Title** **Computer Organization and Architecture Lab**

Course Objectives:

- To provide hands-on understanding of computer system components and their interactions.
- To develop skills in writing and executing ARM/RISC assembly language programs using instruction set simulators.
- To simulate and analyze instruction pipeline behavior, hazards, and execution using architectural simulators such as gem5.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Identify and explain the functional components of a computer system including memory, ports, motherboard, and add-on cards.
- **CO2:** Write and execute ARM/RISC assembly language programs using instruction set simulators.
- **CO3:** Develop C programs to solve computational problems using recursion, sorting techniques, and expression evaluation.
- **CO4:** Analyze instruction pipeline behavior by identifying data hazards, control hazards, and the impact of instruction reordering.
- **CO5:** Configure and use architectural simulators such as gem5 to execute program binaries and observe processor behavior.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	-	-	-	1	-	-	1
CO2	3	3	2	1	2	-	-	-	1	-	-	1
CO3	3	3	3	2	2	-	-	-	1	-	-	1
CO4	3	3	2	3	2	-	-	-	2	-	-	1
CO5	3	2	2	3	3	-	-	-	2	-	-	1

List of Experiments:

1. Computer Anatomy-Memory, Ports, Motherboard, and add-on cards.
2. Write programs in ARM/RISC assembly language and test these on an instruction set simulator.
3. Generate some interesting numbers (example - Happy numbers, Autonomic numbers, Hardy-Ramanujan numbers etc.)
4. Implement a 4-function calculator.
5. Sort an integer array using merge sort (recursive).
6. Evaluate an arithmetic expression specified as a string (using recursive functions).
7. Implement a simple game.
8. Write or generate sequence of instructions and observe the overall pipeline stalls with and without data hazards, control hazards, and with/without data forwarding.
9. Rearrange the sequence of instructions or the program so that the pipeline stalls will be minimized.
10. Configure the simulator [gem5 is preferred] to operate on the binaries of the benchmark as the input.

SEMESTER-IV

Course Code : **BTC401** Course Type : **Core Course** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Design and Analysis of Algorithms**

Course Objectives:

- Understand the fundamental concepts of algorithm analysis and apply mathematical tools to study algorithm efficiency.
- Design and analyze algorithms using various paradigms such as divide-and-conquer, greedy methods, dynamic programming, backtracking, and branch-and-bound.
- Classify computational problems based on complexity classes and understand the concept of NP-completeness and approximation techniques**.**

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Analyze the time and space complexity of algorithms using asymptotic notations and recurrence relations.
- **CO2:** Apply divide-and-conquer and graph traversal techniques to solve problems efficiently.
- **CO3:** Design algorithms using greedy and dynamic programming paradigms to solve optimization problems.
- **CO4:** Apply backtracking and branch & bound approaches and explain NP-completeness and approximation algorithms.
- **CO5:** Classify computational problems into complexity classes such as P, NP, and NP-complete, and explain the role of reductions and approximation algorithms.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	–	–	–	–	–	–	–	–
CO2	3	3	3	2	–	–	–	–	–	–	–	–
CO3	3	3	2	2	–	–	–	–	–	–	–	–
CO4	3	3	2	2	–	–	–	–	–	–	–	–
CO5	3	2	2	2	–	–	–	–	–	–	–	–

Course Outline:

Unit-I: **[09 Hours]**

Introduction to Algorithms: Introduction to algorithms and its importance, asymptotic analysis: worst, average, and best cases, asymptotic notations (Big O, Omega, Theta), complexity analysis of non-recursive algorithms, complexity analysis of recursive algorithms, solution of recurrence relations using substitution method.

Unit-II: **[09 Hours]**

Divide and Conquer: Structure of divide-and-conquer algorithms, binary search, quick sort, merge sort, finding maximum and minimum element, recurrence equations for divide-and-conquer.
Graph algorithms: depth first search (DFS), breadth first search (BFS).

Unit-III: **[09 Hours]**

Greedy Technique: Basics of greedy approach, job sequencing with deadlines, fractional knapsack problem, Huffman coding, minimum cost spanning tree (Prim's, Kruskal's), single source shortest path (Dijkstra's algorithm).Dynamic Programming: overview of dynamic programming, difference between dynamic programming and divide and conquer, matrix chain multiplication, 0/1 knapsack problem

Unit-IV: **[09 Hours]**

Backtracking, Branch & Bound: General method of backtracking, N-Queen problem, 0/1 knapsack using backtracking, general method of branch and bound, traveling salesperson problem (TSP), polynomial-time verification.

Unit-V: **[09 Hours]**

NP-Completeness: NP-completeness and reducibility, NP-completeness proofs, NP-complete problems, introduction to approximation algorithms, tractable and intractable problems, decidable and undecidable problems, reduction, P, NP, NP-complete, Cook's Theorem.

Suggested Readings:

Thomas H. Cormen, Charles E. Leiserson, R.L. Rivest. Introduction to Algorithms, Prentice Hall of India Publications, 3rd Edition 2015.

References:

J. Kleinberg and E. Tardos. Algorithm Design, Pearson 2006.
Sara Baase and Allen Van Gelder. Computer Algorithms: Introduction to Design and Analysis, Pearson education (Singapore) Pvt. Ltd, New Delhi 2007.
Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. The Design and Analysis of Computer Algorithms, Pearson Education (Singapore) 2006.
Algorithmics: Theory and Practice by Brassard and Bratley, Prentice Hall

Course Code : **BTC402** Course Type : **Core Course** No. of Credits : **4.00** No. of Hours : **45T+30L**	**Course Title** **Operating Systems**

Course Objectives:

- To understand the fundamental concepts, evolution, and structural components of operating systems.
- To analyze process management, scheduling algorithms, synchronization, and memory management techniques used in operating systems.
- To evaluate and apply operating system concepts related to file systems, storage management, protection, security, and real-world OS issues.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Understand operating system structures and functionalities.
- **CO2:** Apply and evaluate process scheduling and synchronization techniques.
- **CO3:** Analyze memory and storage management strategies.
- **CO4:** Explain file systems, disk management, and protection mechanisms.
- **CO5:** Analyze deadlock situations and apply prevention, avoidance, detection, and recovery techniques in operating systems.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	–	–	–
CO2	3	3	2	2	–	–	–	–	–	–	–	–
CO3	3	3	2	2	–	–	–	–	–	–	–	–
CO4	3	2	–	–	–	–	–	–	–	–	–	–
CO5	3	3	2	2	–	–	–	–	–	–	–	–

Course Outline:

Unit-I: **[15 Hours]**

Introduction and System Structures: Evolution of operating systems, Batch, Multi-programming, Time sharing, Multiprocessor, Distributed, Cluster, Real-time systems, Linux system introduction and commands, Computer system structure, Network structure, I/O structure, Storage structure, System components, Operating-system services, System calls, Types of system calls, System programs, System structure, Virtual machines, System design and implementation.

Unit-II: **[15 Hours]**

Process Management and CPU Scheduling: Process concept, Process control block and attributes, Process states and multi-programming, Process state transition diagram, Various process schedulers, Threads, Scheduling criteria, First Come First Serve, Shortest Job First, Shortest Remaining Time First, Round Robin scheduling algorithm, Longest Job First, Longest Remaining Time First, Highest Response Ratio Next, Priority scheduling (pre-emptive and non-pre-emptive), Multilevel queue scheduling, Multilevel feedback queue scheduling.

Unit-III: **[15 Hours]**

Process Synchronization and Deadlocks: Synchronization background, Race condition, Critical section problem: Mutual exclusion, Progress, Bounded waiting, Peterson's solution, Semaphores: Wait, Signal, Binary semaphores, Semaphores and monitors, Producer-consumer problem, Reader-writer problem, Dining philosopher's problem, Deadlock introduction, Resource allocation graphs: Single instance, Multiple instance, Necessary conditions for deadlock, Deadlock handling methods, Deadlock prevention, Deadlock avoidance: Banker's algorithm, Deadlock detection and recovery.

Unit-IV: **[15 Hours]**

Memory Management: Memory management background, Swapping, Logical and physical address space, Contiguous memory allocation: First fit, Best fit, Worst fit, Internal and external fragmentation, Non-contiguous memory allocation: Paging, Structure of the page table, Multilevel paging, Inverted paging, Segmentation, Segmentation with paging, Address translations, Virtual memory, Demand paging, Page replacement algorithms, Allocation of frames, Thrashing, Memory-mapped I/O.

Unit-V: **[15 Hours]**

File, and Disk Management: File concept, File attributes, File access methods, Directory structure, File system organization and mounting, File sharing, Protection, File-system structure, File-system implementation, Directory implementation, Allocation methods, Free-space management, Efficiency and performance, Disk structure, Logical and physical view, Disk scheduling algorithms: First Come First Serve, Shortest Seek Time First, SCAN, C-SCAN, LOOK, C-LOOK, Protection and security**.**

Suggested Readings:

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles, Wiley, 8/e.

References:

W. Stallings, Operating Systems Internals and Design Principles, Prentice Hall (2013) 7th ed.
Andrew S. Tanenbaum, Operating Systems: Design and Implementation, Pearson (2006)
Andrew S. Tanenbaum, Modern Operating Systems, Pearson 5th edition (2022)
Dhamdhere, D.M., Operating Systems: A Concept Based Approach, McGraw Hill (2008) 2nd ed.
Flynn, I.M. and McHoes, A.M., Understanding Operating Systems, Thomson (2007).
C. Crowley, Operating Systems. A Design-Oriented Approach, Tata McGraw-Hill Publishing
Company Limited, 1998.
C. Crowley, Operating Systems. A Design-Oriented Approach, Tata McGraw-Hill Publishing Company Limited, 1998.

Course Code : **BTC403** Course Type : **Core Course** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Database Management Systems**

Course Objectives:

- To understand basic concepts of database systems, data models, and database design.
- To design normalized relational databases and write queries using SQL and relational algebra with proper integrity constraints.
- To explain and apply transaction processing, query handling, and recovery techniques in database systems.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Understand the fundamentals of database systems, DBMS architecture, data models, and conceptual database design using the ER model.
- **CO2:** Formulate queries using relational algebra, relational calculus, and SQL to retrieve and manipulate data from relational databases.
- **CO3:** Design efficient relational database schemas by applying normalization techniques and dependency analysis.
- **CO4:** Analyze transaction management, concurrency control, and deadlock handling mechanisms to ensure database consistency and reliability.
- **CO5:** Explain query processing, optimization techniques, and recovery mechanisms used in database systems.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	–	–	2
CO2	3	3	2	–	2	–	–	–	–	–	–	2
CO3	3	3	3	2	–	–	–	–	–	–	–	2
CO4	3	3	2	2	–	–	–	–	–	–	–	2
CO5	3	3	2	2	–	–	–	–	–	–	–	2

Course Outline:

Unit-I: **[08 Hours]**

Introduction to DBMS and Data Models: Historical perspective, File versus a DBMS, Advantages of DBMS, Describing and storing data in DBMS, Architecture of a DBMS, Different data models, Features of ER model, Conceptual design using ER model, Design for large enterprises, Relational model – structure and operations, Integrity constraints over relations.

Unit-II: **[10 Hours]**

Query Languages and SQL: Relational Algebra, Relational Calculus, SQL – Queries, Constraints, Form of SQL query, UNION, INTERSECT and EXCEPT, Nested queries, Aggregate operators, Null values, Complex integrity constraints in SQL, Triggers, Embedded SQL.

Unit-III: **[09 Hours]**

Database Design and Normalization: Mapping ER model to relational form, Functional dependency – Closure of functional dependencies, Closure of attributes, Canonical cover, Properties of decompositions, Normalization process – 1NF, 2NF, 3NF, BCNF, Multivalued dependency – Closure and properties, 4NF, Join dependency – PJNF, Decomposition algorithms.

Unit-IV: **[09 Hours]**

Transaction Management and Concurrency Control: ACID properties, Transactions, Schedules and concurrent execution, Concurrency control – Lock-based protocol, Serializability, Recoverability, Dealing with deadlocks, Concurrency control without locking.

Unit-V: **[09 Hours]**

Query Processing and Recovery: Overview of query evaluation, Operator evaluation, Algorithms for relational operations – Selection, Projection, Join, Set operations, Aggregate operations, Evaluation of relational operations, Query optimization – Alternative plans, Query optimizer functions, Translating SQL to relational algebra, Cost estimation, Relational algebra equivalences, Other optimization techniques, Failure classification, Recovery and atomicity, Log-based recovery, Shadow paging, Advanced recovery techniques.

Suggested Readings:

Silberschatz, Korth and Sudarshan, "Database System Concepts", McGraw Hill, 2021, 7th Edition.

References:

Elamsri and Navathe, "Fundamentals of Database Systems", Pearson Education, 2017,7th Edition.

Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", McGraw Hill, 2002

Course Code : **BTC404** Course Type : **Core Course** No. of Credits : **4.00** No. of Hours : **45 T + 30 L**	**Course Title** **Web Technologies**

Course Objectives:

- To understand fundamental web concepts including client-server architecture, networking basics, and web protocols.
- To design and develop interactive web pages using HTML, CSS, and JavaScript.
- To implement server-side programming using PHP, Servlets, and JDBC with database integration.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Design and structure basic web pages using HTML and style them using CSS.
- **CO2:** Implement dynamic functionality on web pages using JavaScript for client-side processing.
- **CO3:** Develop interactive web applications using PHP with file handling, forms, and database operations.
- **CO4:** Build server-side components using Servlets and JDBC for database-driven applications.
- **CO5:** Apply concepts of web protocols, client-server architecture, sessions, and cookies to design secure and efficient web applications.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	3	-	2	-	-	-	1	2	-	-
CO2	2	2	3	1	2	-	-	-	1	2	-	-
CO3	3	2	3	2	3	-	-	-	1	2	1	-
CO4	3	3	3	3	3	-	-	-	2	2	2	1
CO5	3	3	2	2	2	-	-	-	1	2	1	1

Course Outline:

Unit-I: **[15 Hours]**

Introduction to Web and Design Fundamentals: Internet overview, fundamental computer network concepts, web protocols, URL, domain name, web browsers and web servers, working principle of a website.

Unit-II: **[15 Hours]**

Website Design and Client-Side Basics: creating a website, client-side and server-side scripting, HTML form elements, input types, media elements, CSS3 selectors, box model, backgrounds and borders, text effects, animations, multiple column layout, user interface.

Unit-III: **[15 Hours]**

Client-Side Scripting with JavaScript: JavaScript introduction, variables and data types, statements, operators, literals, functions, objects, arrays, built-in objects, regular expressions, exceptions, event handling, validation, JavaScript debuggers.

Unit-IV: **[15 Hours]**

Server-Side Scripting with PHP: Working principle of PHP, PHP variables, constants, operators, flow control and looping, arrays, strings, functions, file handling, file uploading, email basics, email with attachments, PHP and HTML integration, simple PHP scripts, databases with PHP**.**

Unit-V: **[15 Hours]**

Advanced Server-Side: Servlets and JDBC: Java servlet architecture, servlet life cycle, form GET and POST actions, sessions, cookies, database connectivity, JDBC, creation of simple interactive applications, simple database applications.

Suggested Readings:

Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5", O'Reilly publishers, 3rd *Edition*, 2014.
Paul Deitel, Harvey Deitel, Abbey Deitel, "Internet & World Wide Web - How to Program," Pearson Education, 5th *Edition*, 2012.

References:

Jeffrey C. Jackson, "Web Technologies-A Computer Science Perspective," Pearson Education, 2006.
James F. Kurose, "Computer Networking: A Top-Down Approach", Pearson Education, 6th Edition, 2012.
Steven Holzener, "PHP – The Complete Reference", Mc-Graw Hill, 1st Edition, 2017.
Fritz Schneider, Thomas Powell, "JavaScript – The Complete Reference", McGraw Hill Publishers, 3rd *Edition*, 2017.

Course Code : **BTC405** Course Type : **Core Course** No. of Credits : **3.00** No. of Hours : 45	**Course Title** **Theory of Computation**

Course Objectives:

- To introduce the fundamental concepts of automata theory, formal languages, and models of computation.
- To understand and design finite automata, regular expressions, and context-free grammars for language representation.
- To study advanced computational models such as pushdown automata and Turing machines, including decidability and undecidability concepts.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Understand the foundational principles of automata theory and formal proof techniques.
- **CO2:** Design deterministic and non-deterministic finite automata to recognize regular languages.
- **CO3**: Apply regular expressions and algebraic laws to represent and manipulate regular languages.
- **CO4:** Construct context-free grammars and pushdown automata to describe context-free languages.
- **CO5:** Analyze the capabilities and limitations of Turing machines, including decidability and undecidability problems.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	1	-	-
CO2	3	3	3	1	2	-	-	-	1	2	-	-
CO3	3	3	2	2	2	-	-	-	1	2	-	-
CO4	3	3	3	2	3	-	-	-	2	2	1	-
CO5	3	3	2	3	2	-	-	-	1	2	1	1

Course Outline:

Unit-I: **[08 Hours]**

Automata: The Methods and the Madness – Why study automata theory – Introduction to formal proof – Additional forms of proof – Inductive proofs – The central concepts of automata theory. **Finite Automata** – An informal picture of finite automata – Deterministic finite automata – Nondeterministic finite automata – An application: text search – Finite automata with epsilon-transitions.

Unit-II: **[09 Hours]**

Regular Expressions and Languages – Regular expressions – Finite automata and regular expressions – Applications of regular expressions – Algebraic laws for regular expressions. **Properties of Regular Languages** – Proving languages not to be regular – Closure properties of regular languages – Decision properties of regular languages – Equivalence and minimization of automata.

Unit-III: **[08 Hours]**

Context-Free Grammars and Languages – Context-free grammars – Parse trees – Applications of context-free grammars – Ambiguity in grammars and languages.

Unit-IV: **[10 Hours]**

Pushdown Automata – Definition of pushdown automata – The languages of a PDA – Equivalence of PDA's and CFG's – Deterministic pushdown automata.
Properties of CFLs – Pumping lemma – Closure – Decision properties**.**

Unit-V: **[12 Hours]**

Introduction to Turing Machines: Problems that computers cannot solve; The Turing machine; Programming techniques for Turing machines; Extensions to the basic Turing machine; Restricted Turing machine; Turing machines and computers.
Undecidability: A language that is not recursively enumerable; An undecidable problem that is RE; Undecidable problems about Turing machines; Post's correspondence problem; Other undecidable problems.

Suggested Readings:

John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education, 2006, Third Edition.

References:

Peter Linz, An Introduction to Formal Languages and Automata, Jones, and Bartlett Learning, 2013, Sixth Edition.
K. L. P. Mishra and N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI, 2006, Third Edition.

Course Code : **BTC406** Course Type : **Skill Course** No. of Credits **: 2.00** No. of Hours : **45**	**Course Title** **Power BI**

Course Objectives:

- To introduce basic concepts of Business Intelligence and data visualization.
- To understand data handling and preparation techniques.
- To learn simple data modeling and calculations.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Understand basic concepts of BI and Power BI tools
- **CO2:** Perform data loading and basic data preparation
- **CO3:** Apply simple data modeling and calculations
- **CO4:** Create basic visualizations and reports
- **CO5:** Publish reports and understand practical applications

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	–	–	–	–	–	–	1	–	2
CO2	3	3	2	–	–	–	–	–	1	1	–	2
CO3	3	3	2	1	–	–	–	–	1	1	–	2
CO4	3	2	3	2	1	–	–	–	2	2	–	2
CO5	2	2	2	2	3	–	–	–	2	2	1	3

Course Outline:

Unit-I: **[08 Hours]**

Basics of Power BI: Introduction to BI and data visualization, Overview of Power BI tools, Installation and interface, Basic dashboard creation

Unit-II: **[09 Hours]**

Data Handling: Connecting to different data sources (Excel, CSV, Database), Data Import vs Direct Query, Data transformation using Power Query, Data cleaning: handling missing values, filtering, sorting, Data merging and shaping

Unit-III: **[08 Hours]**

Data Modeling: Introduction to data models, Creating relationships between tables, Calculated columns and measures, Basics of DAX functions, Simple time-based analysis

Unit-IV: **[10 Hours]**

Communication Visualization: Charts and reports, Filters and interactivity, Custom visuals and formatting, Basics of forecasting and trend analysis

Unit-V: **[12 Hours]**

Sharing and Applications: Publishing and sharing reports, Basic security, Performance basics, Practical applications

Suggested Readings:

Microsoft Power BI Step by Step – Brett Powell
Beginning Microsoft Power BI – Dan Clark
Pro Power BI Desktop – Adam Aspin

References:

Official Microsoft Documentation: https://learn.microsoft.com/power-bi/
Power BI Learning (Microsoft Learn): https://learn.microsoft.com/training/powerplatform/power-bi
Tutorials Point Power BI Guide: https://www.tutorialspoint.com/power_bi
Guru99 Power BI Tutorial: https://www.guru99.com/power-bi-tutorial.html

Course Code : **BTC407** Course Type : **Core Lab** No. of Credits : **2.00** No. of Hours : **60**	**Course Title** **Design and Analysis of Algorithms Lab**

Course Objectives:

- To implement and analyze fundamental algorithms using appropriate design techniques such as divide and conquer, greedy, and dynamic programming.
- To develop problem-solving skills by designing efficient algorithms and evaluating their time complexity.
- To apply algorithmic strategies like backtracking and graph algorithms for solving real-world computational problems.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Implement basic algorithms and evaluate their time and space complexity.
- **CO2:** Design and implement divide and conquer algorithms such as sorting and searching techniques.
- **CO3:** Apply greedy and dynamic programming approaches to solve optimization problems.
- **CO4:** Implement graph algorithms such as BFS, DFS, minimum spanning tree, and shortest path.
- **CO5:** Analyze algorithm efficiency and compare different approaches for solving a given problem.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	3	-	-	-	1	2	-	-
CO2	3	3	3	2	3	-	-	-	1	2	-	-
CO3	3	3	3	2	3	-	-	-	1	2	-	-
CO4	3	3	3	2	3	-	-	-	2	2	-	-
CO5	3	3	2	3	3	-	-	-	1	2	1	1

Course Outline:

List of Experiments:

1. Analyze time complexity of linear search and binary search (best, average, worst case).
2. Compare iterative and recursive implementations (e.g., factorial) and analyze complexity.
3. Solve recurrence relations using substitution method (verification program).
4. Implement Binary Search using divide and conquer.
5. Implement Merge Sort.
6. Implement Quick Sort.
7. Find maximum and minimum elements using divide and conquer.
8. Implement Breadth First Search (BFS).
9. Implement Depth First Search (DFS).
10. Solve Fractional Knapsack problem using greedy method.
11. Implement Huffman Coding.
12. Implement Minimum Spanning Tree using Prim's algorithm.
13. Implement Minimum Spanning Tree using Kruskal's algorithm.
14. Implement Dijkstra's algorithm for shortest path.
15. Solve 0/1 Knapsack problem using dynamic programming.

Suggested Readings:

Thomas H. Cormen, Charles E. Leiserson, R.L. Rivest. Introduction to Algorithms, Prentice Hall of India Publications, 3rd Edition 2015.

References:

J. Kleinberg and E. Tardos. Algorithm Design, Pearson 2006.
Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. The Design and Analysis of Computer Algorithms, Pearson Education (Singapore) 2006.
Algorithmics: Theory and Practice by Brassard and Bratley, Prentice Hall

Course Code : **BTC408** Course Type : **Core Lab** No. of Credits : **2.00** No. of Hours : **60**	**Course Title** **Database Management Systems Lab**

Course Objectives:

- To design and implement database schemas using SQL, including ER modeling, relational mapping, and integrity constraints.
- To develop skills in writing SQL queries for data manipulation, retrieval, and analysis using joins, aggregates, and subqueries.
- To understand and implement advanced database concepts such as transactions, normalization, indexing, and triggers.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Create and manage databases using SQL DDL and DML commands with appropriate constraints.
- **CO2:** Design ER models and convert them into normalized relational schemas up to BCNF.
- **CO3:** Write efficient SQL queries using joins, aggregate functions, subqueries, and set operations.
- **CO4:** Implement transaction management, concurrency control, and recovery concepts using SQL commands.
- **CO5:** Apply advanced database features such as views, indexing, triggers, and stored procedures for real-world applications.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	3	-	-	-	1	2	-	-
CO2	3	3	3	1	2	-	-	-	1	2	-	-
CO3	3	3	3	1	3	-	-	-	1	2	-	-
CO4	3	3	2	2	2	-	-	-	1	2	-	-
CO5	3	2	3	1	3	-	-	-	2	2	-	-

List of Experiments:

1. Create a database and tables using SQL DDL commands (CREATE, ALTER, DROP).
2. Design an ER model for a real-world application (such as Student or Library system) and convert it into relational tables.
3. Implement integrity constraints such as Primary Key, Foreign Key, NOT NULL, UNIQUE, and CHECK in table creation.
4. Perform data manipulation operations (INSERT, UPDATE, DELETE, SELECT) on multiple tables.
5. Write SQL queries using WHERE, ORDER BY, GROUP BY, and HAVING clauses.
6. Write SQL queries using aggregate functions such as SUM, AVG, COUNT, MIN, and MAX.
7. Implement different types of JOIN operations (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN).
8. Perform set operations using SQL such as UNION, INTERSECT, and EXCEPT.
9. Write SQL queries using nested queries and subqueries (IN, EXISTS, ANY, ALL).
10. Create and use views for data abstraction and security.
11. Perform normalization of a given relation up to BCNF.
12. Write SQL statements to demonstrate transaction control commands such as COMMIT, ROLLBACK, and SAVEPOINT.
13. Demonstrate concurrency control concepts such as serializability, locking, and deadlocks through SQL-based examples.
14. Create indexes on tables and analyze their impact on query performance.
15. Implement triggers and stored procedures for automatic operations in the database.

Suggested Readings:

Silberschatz, Korth and Sudarshan, *"Database System Concepts"*, McGraw Hill, 2021, 7th Edition

References:

Elmasri and Navathe, *"Fundamentals of Database Systems"*, Pearson Education, 2017, 7th Edition.
Raghu Ramakrishnan, Johannes Gehrke, *"Database Management Systems"*, McGraw Hill, 2002.
Alan Beaulieu, *"Learning SQL"*, O'Reilly Media, 2nd Edition 2020.
James R. Groff and Paul N. Weinberg, *"SQL: The Complete Reference"*, McGraw Hill, 3rd Edition.

Course Code : **BTC409** Course Type : **Mandatory** No. of Credits : **0.00** No. of Hours : **30**	**Course Title** **Environmental Sciences**

Course Objectives:

- To understand the fundamental concepts of ecosystems, biodiversity, and natural resources.
- To analyze environmental pollution, its impacts, and control technologies along with global environmental issues.
- To study environmental policies, legislation, and sustainable development practices for environmental protection.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain the structure and functioning of ecosystems, energy flow, and biogeochemical cycles.
- **CO2:** Analyze the utilization and conservation of natural resources and biodiversity.
- **CO3:** Identify different types of environmental pollution and evaluate appropriate control measures and technologies.
- **CO4:** Understand environmental laws, policies, and the process of Environmental Impact Assessment (EIA).
- **CO5:** Apply principles of sustainable development and environmental management in real-world scenarios.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	-	-	-	3	3	2	-	1	-	-
CO2	2	2	1	-	-	3	3	2	-	1	-	-
CO3	2	2	1	1	1	3	3	2	-	1	-	-
CO4	2	1	-	-	-	3	3	3	-	2	-	-
CO5	2	2	2	1	1	3	3	3	1	2	-	1

Course Outline:

Unit-I: **[06 Hours]**

Ecosystems: Definition, Scope and Importance of ecosystem. Classification, structure and function of an ecosystem, Food chains, food web and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Bio magnification, ecosystem value, services and carrying capacity.

Unit-II: **[06 Hours]**

Natural Resources: Classification of Resources: Living and Non-Living resources, water resources: use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources, Land resources: Forest resources, Energy resources: growing energy needs, renewable and non renewable energy sources, use of alternate energy source, case studies.

Unit-III: **[06 Hours]**

Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Exsitu conservation. National Biodiversity act.

Unit-IV: **[06 Hours]**

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, Air Pollution: Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. Water pollution: Sources and types of pollution, drinking water quality standards. Soil Pollution: Sources and types, Impacts of modern agriculture,. Noise Pollution: Sources and Health hazards, standards, Solid waste: Municipal Solid Waste management, composition and characteristics of e-Waste and its management. Pollution control technologies: Wastewater Treatment methods: Primary, secondary andTertiary. Overview of air pollution control technologies, Concepts of bioremediation. Global Environmental Problems and Global Efforts: Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS).. International conventions / Protocols: Earth summit, Kyoto protocol and Montréal Protocol.

Unit-V: **[06 Hours]**

Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio-economical aspects. Strategies for risk assessment, Towards Sustainable Future: Concept of Sustainable Development, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

Suggested Readings:

Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
Environmental Studies by R. Rajagopalan, Oxford University Press.

References:

Environmental Science: towards a sustainable future by Richard T.Wright. 2008 PHL Learning Private Ltd. New Delhi.
Environmental Engineering and Science by Gilbert M.Masters and Wendell P. Ela .2008 PHI Learning Pvt. Ltd.
Environmental Science by Daniel B.Botkin & Edward A.Keller, Wiley INDIA edition.
Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.

SEMESTER-V

Course Code : **BTC501** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Artificial Intelligence and Machine Learning**

Course Objectives:
- To understand and analyze the fundamental concepts of Artificial Intelligence and intelligent agents.
- To apply machine learning techniques for solving real-world problems.
- To analyze and implement AI and ML methods in practical applications.

Course Outcomes:

After completion of the course, students will be able to
- **CO1:** Explain the concepts of intelligent agents and problem-solving techniques.
- **CO2:** Apply search algorithms to solve AI problems.
- **CO3:** Analyze knowledge representation and reasoning methods.
- **CO4:** Apply machine learning algorithms for classification and clustering tasks.
- **CO5:** Analyze and implement AI-based applications using advanced techniques.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	–	–	1
CO3	3	3	–	2	–	–	–	–	–	–	–	1
CO4	3	3	2	2	2	–	–	–	–	–	–	1
CO5	3	3	2	2	2	–	–	1	1	1	–	2

Course Outline:

UNIT-I **[07 hours]**

Introduction to Artificial Intelligence: Definition, history, applications, intelligent agents, PEAS description, environment types, problem formulation, state space representation, production systems.

UNIT-II **[10 hours]**

Search and Optimization Techniques: Uninformed search – BFS, DFS, Uniform Cost Search; Informed search – Greedy Best First Search, A* algorithm; Adversarial search – Minimax, Alpha-Beta pruning; Local search – Hill Climbing, Simulated Annealing; Evolutionary computation – Genetic algorithms.

UNIT-III **[10 hours]**

Knowledge Representation: Propositional logic, First Order Predicate Logic, inference mechanisms – forward and backward chaining, resolution, semantic networks, frames, ontologies, probabilistic reasoning, Bayesian networks.

UNIT-IV **[10 hours]**

Machine Learning Techniques: Introduction to machine learning, types of learning – supervised, unsupervised, reinforcement learning; Regression – linear regression; Classification – decision trees, K-nearest neighbors, Naïve Bayes; Clustering – K-means; Model evaluation metrics – accuracy, precision, recall, F1-score.

UNIT-V **[08 hours]**

Advanced AI and ML: Neural networks, deep learning, reinforcement learning concepts, natural language processing, computer vision, AI applications, ethical issues in AI.

Suggested Readings:

Stuart Russell and Peter Norvig, "Artificial Intelligence A Modern Approach", Pearson Education, 2020, Fourth Edition.
Tom M. Mitchell, "Machine Learning", McGraw-Hill Education, 1997, First Edition.

References:

C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
E. Rich and K. Knight, *Artificial Intelligence*, 3rd ed. New York, NY, USA: McGraw-Hill, 2009.
Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

Course Code : **BTC502** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Computer Networks**

Course Objectives:

- To analyze the fundamental concepts of computer networks, including network architectures, reference models, and protocol design.
- To apply data link, network, and transport layer protocols for efficient data communication and routing.
- To analyze and evaluate network services, application layer protocols, and basic network security mechanisms.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain and analyze OSI and TCP/IP reference models.
- **CO2:** Analyze MAC layer protocols and LAN technologies.
- **CO3:** Design network applications using Internet protocols.
- **CO4:** Apply and implement routing and congestion control algorithms.
- **CO5:** Develop and evaluate application layer protocols and services.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	1	–	1
CO2	3	3	–	2	1	–	–	–	–	–	–	1
CO3	3	2	3	2	2	–	–	–	1	1	–	1
CO4	3	3	2	2	2	–	–	–	–	–	–	1
CO5	3	3	2	2	1	–	–	1	1	1	–	2

Course Outline:

UNIT-I **[07 hours]**

Introduction – network architecture - protocol implementation issues - network design. Reference models- The OSI Reference Model- the TCP/IP Model - A Comparison of the OSI and TCP/IP Models

UNIT-II **[10 hours]**

Datalink Layer-Ethernet, Token ring, wireless LANs-Issues with data link Protocols-Encoding framing and error detection and correction-sliding window Protocol-Medium access control

UNIT-III **[10 hours]**

Network layer – network layer design issues - Routing algorithms - Congestion control algorithms – Internetworking - The network layer in the internet - Internet Protocol (IP) - Unicast, multicast, and inter domain routing

UNIT-IV **[09 hours]**

Transport layer - Elements of transport protocol - Congestion control – The Internet's Transmission Control Protocol (TCP) - Remote Procedure Call (RPC) – Implementation semantics of RPC – BSD sockets - client-server applications.

UNIT-V **[09 hours]**

Application layer - Domain name server – Simple Mail Transfer Protocol – File Transfer Protocol - World wide web - Hypertext transfer protocol -Presentation formatting and data compression- Introduction to Network security - Web Services architectures for developing new application protocols.

Suggested Readings:

S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 6th ed. Hoboken, NJ, USA: Pearson, 2021.

J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 8th ed. Boston, MA, USA: Pearson, 2021.

References:

L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2021.

W. Stallings, *Data and Computer Communications*, 10th ed. Upper Saddle River, NJ, USA: Pearson, 2013.

K. R. Fall and W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2011.

N. Olifer and V. Olifer, *Computer Networks: Principles, Technologies and Protocols for Network Design*, 2nd ed. Hoboken, NJ, USA: Wiley, 2006.

Course Code : **BTC503** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Compiler Design**

Course Objectives:

- To analyze the fundamental phases of a compiler, including lexical, syntax, and semantic analysis.
- To apply parsing techniques and syntax-directed translation methods for processing programming languages.
- To design and evaluate intermediate code generation and optimization techniques for efficient program execution.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain and analyze the structure of a compiler and lexical analysis techniques using finite automata and regular expressions.
- **CO2:** Apply parsing techniques to construct syntax trees and implement different parsing algorithms.
- **CO3:** Analyze and implement syntax-directed translation schemes and symbol table management.
- **CO4:** Apply code generation and storage allocation techniques for intermediate code representation.
- **CO5:** Analyze and evaluate code optimization methods and error handling mechanisms in compilers

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	1	–	1
CO2	3	3	2	2	1	–	–	–	–	–	–	1
CO3	3	3	2	2	2	–	–	–	–	–	–	1
CO4	3	2	3	2	2	–	–	–	–	–	–	1
CO5	3	3	2	2	1	–	–	1	1	1	–	2

Course Outline:

UNIT-I **[09 hours]**

Introduction & Lexical Analysis: Phases of compiler, passes, bootstrapping, finite state machines, regular expressions and their applications to lexical analysis, lexical analyzer generator, LEX compiler.

UNIT-II **[09 hours]**

Syntax Analysis Basics: Formal grammars and their application to syntax analysis, ambiguity, context free grammars, derivations, parse trees, top-down parsing, predictive parsers.

UNIT-III **[09 hours]**

Advanced Parsing: Shift-reduce parsing, operator precedence parsing, construction of SLR parsing tables, canonical LR parsing tables, LALR parsing tables, automatic parser generator, YACC

UNIT-IV **[09 hours]**

Syntax Directed Translation & Storage: Syntax-directed translation, translation of assignment statements, Boolean expressions, postfix translation, translation with top-down parser, symbol tables, storage allocation, stack allocation, block structured languages.

UNIT-V **[09 hours]**

Error Handling & Code Optimization: Error detection and recovery, lexical errors, syntactic errors, semantic errors, code generation, machine-independent code optimization, loop optimization, DAG representation of basic blocks.

Suggested Readings:

V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, Updated 2nd ed. New Delhi, India: Pearson, 2023.

K. Muneeswaran, *Compiler Design*, 1st ed. New Delhi, India: Oxford University Press, 2012.

References:

K. D. Cooper and L. Torczon, *Engineering a Compiler*, 3rd ed. Cambridge, MA, USA: Morgan Kaufmann, 2022.

T. Æ. Mogensen, *Introduction to Compiler Design*, 3rd ed. Cham, Switzerland: Springer, 2024.

K. C. Louden, *Compiler Construction: Principles and Practice*. Boston, MA, USA: Cengage Learning, 1997.

Course Code : **BTC504** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Information Retrieval**

Course Objectives:

- To understand the fundamentals, architecture, and applications of Information Retrieval systems
- To analyze query processing and text representation methods
- To apply classification and clustering techniques for text data

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain IR concepts, architecture, and retrieval models
- **CO2:** Apply indexing and ranking techniques
- **CO3:** Analyze query processing and text representation
- **CO4**: Apply classification and clustering techniques
- **CO5**: Analyze web-based IR and information extraction

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	3	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	2	3	3	–	–	–	1	1	–	1
CO5	3	3	2	3	3	1	–	1	1	1	–	2

Course Outline:

UNIT I **[09 hours]**

Introduction to Information Retrieval: Goals and History of Information Retrieval, Information Retrieval vs Data Retrieval, Impact of Web on IR **,** Role of Artificial Intelligence in IR, Components of IR Systems, Search Engine Architecture.

UNIT II **[09 hours]**

Retrieval Models and Indexing: Boolean Retrieval Model, Vector Space Model, Ranked Retrieval, Text Similarity Metrics, TF-IDF Weighting, Cosine Similarity, Tokenization, Stop-word Removal, Stemming, Inverted Index Construction.

UNIT III **[09 hours]**

Query Processing and Text Representation: Query Operations, Query Languages, Relevance Feedback, Query Expansion, Word Statistics, Zipf's Law, Porter Stemmer, Morphological Processing, Index Term Selection, Metadata and Markup Languages, SGML, HTML and XML.

UNIT IV **[09 hours]**

Text Categorization and Clustering: Text Categorization, Naive Bayes Classification, Decision Trees, Nearest Neighbor Classification, Clustering Techniques, Agglomerative Clustering, K-Means Clustering, Expectation Maximization (EM), Applications in Information Filtering and Organization.

UNIT V **[09 hours]**

Information Extraction and Integration: Information Extraction from Text, Web Information Extraction, XML-based Information Processing, Semantic Web Concepts, Information Integration Techniques, Specialized Search Systems, Web Data Integration, Recent Trends in Information Retrieval.

Suggested Readings:

Introduction to Information Retrieval – Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Cambridge University Press
Modern Information Retrieval: The Concepts and Technology behind Search – Addison-Wesley
Information Retrieval: Algorithms and Heuristics – Springer
Search Engines: Information Retrieval in Practice – Addison-Wesley .

References:

Mining of Massive Datasets – Cambridge University Press
Speech and Language Processing – Pearson
Web Data Management – Cambridge University Press
Pattern Recognition and Machine Learning – Springer

Course Code : **BTC504** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Cloud Computing**

Course Objectives:

- To understand cloud computing concepts, models, and architecture
- To analyze cloud platforms and programming environments
- To apply cloud computing solutions in real-world applications.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain cloud concepts and virtualization
- **CO2:** Apply cloud service models
- **CO3:** Analyze cloud programming platforms
- **CO4:** Evaluate cloud security and reliability
- **CO5:** Apply cloud solutions in real scenarios

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	3	–	–	–	–	1	–	1
CO3	3	3	3	3	3	–	–	–	1	1	–	1
CO4	3	3	2	3	3	1	–	1	–	1	–	2
CO5	3	3	3	3	3	1	–	1	1	1	–	2

Course Outline:

UNIT I **[09 hours]**

History of Centralized and Distributed Computing - Overview of Distributed Computing, Cluster computing, Grid computing. Technologies for Network based systems- System models for Distributed and cloud computing- Software environments for distributed systems and clouds.

UNIT II **[09 hours]**

Introduction to Cloud Computing- Cloud issues and challenges - Properties - Characteristics - Service models, Deployment models. Cloud resources: Network and API - Virtual and Physical computational resources - Data-storage. Virtualization concepts - Types of Virtualization- Introduction to Various Hypervisors - High Availability (HA)/Disaster Recovery (DR) using Virtualization, Moving VMs .

UNIT III **[09 hours]**

Service models - Infrastructure as a Service (IaaS) - Resource Virtualization: Server, Storage, Network - Case studies. Platform as a Service (PaaS) - Cloud platform & Management: Computation, Storage - Case studies. Software as a Service (SaaS) - Web services - Web 2.0 - Web OS - Case studies – Anything as a service (XaaS).

UNIT IV **[09 hours]**

Cloud Programming and Software Environments – Parallel and Distributed Programming paradigms – Programming on Amazon AWS and Microsoft Azure – Programming support of Google App Engine – Emerging Cloud software Environment.

UNIT V **[09 hours]**

Cloud Access: authentication, authorization and accounting - Cloud Provenance and meta-data - Cloud Reliability and fault-tolerance - Cloud Security, privacy, policy and compliance- Cloud federation, interoperability and standards.

Suggested Readings:

Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, “Distributed and cloud computing from Parallel Processing to the Internet of Things”, Morgan Kaufmann, Elsevier – 2012

References:

Barrie Sosinsky, “ Cloud Computing Bible” John Wiley & Sons, 2010

Tim Mather, Subra Kumaraswamy, and Shahed Latif, Cloud Security and Privacy An Enterprise Perspective on Risks and Compliance, O'Reilly 2009

Course Code : **BTC504** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Algorithms & Data Structures**

Course Objectives:

- To understand advanced data structures and algorithm complexity
- To analyze sorting and searching techniques
- To apply algorithm design paradigms

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Analyze complexity and basic structures
- **CO2:** Apply trees and hashing
- **CO3:** Analyze graph algorithms
- **CO4:** Apply sorting/searching techniques
- **CO5:** Design solutions using algorithm paradigms

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	2	2	2	–	–	–	–	1	–	1
CO5	3	3	3	3	2	–	–	–	1	1	–	2

Course Outline:

UNIT I **[09 hours]**

Review of order rotation & growth of functions, recurrences, probability distributions, Average case analysis of algorithms, Basic data structures such as stacks, queues, linked lists, and applications.

UNIT II **[09 hours]**

Direct access tables and hash tables, hash functions and relates analysis, Binary Search trees and Operations, AVL Trees and balancing operations, R B Trees, properties, operations.

UNIT III **[09 hours]**

B – Trees – definition – properties, operations, data structures for disjoint sets, Graph algorithms, MST single source all pair shortest paths, BFS, DFS, topological sort, strongly connected components.

UNIT IV **[09 hours]**

Quick sort randomized version, searching in linear time, More graph algorithms – maximal independent sets, coloring vertex cover, introduction to perfect graphs.

UNIT V **[09 hours]**

Algorithmic paradigms Greedy Strategy, Dynamic programming, Backtracking, Branch-and-Bound, Randomized algorithms.

Suggested Readings:

H. S. Wilf, Algorithms and complexity, Prentice hall.

References:

T. H. Cormen, C. E. Leiserson, R. L. Rivest, Introduction to Algorithms, Prentice hall.
K. Vishwanathan Iyer, Lecture notes for classroom use.

Course Code : **BTC504** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Human-computer interaction**

Course Objectives:

- To understand principles and design of human-computer interaction
- To learn user-centered design and evaluation techniques
- To evaluate interfaces for real-world applications

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand HCI principles
- **CO2:** Apply user research methods
- **CO3:** Analyze usability and cognition
- **CO4:** Design and evaluate interfaces
- **CO5:** Apply HCI in real-world systems

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	–	–	–	–	2	–	1
CO2	3	3	2	2	2	1	–	1	1	2	–	1
CO3	3	3	2	3	1	–	–	–	–	2	–	1
CO4	3	3	3	3	2	1	–	1	1	2	–	2
CO5	3	3	3	3	2	1	–	1	1	2	–	2

Course Outline:

UNIT I **[09 hours]**

Introduction to HCI. What is HCI – Interdisciplinary nature of HCI – History of HCI – Importance of design and HCI – The design process -- The double diamond of design.

UNIT II **[09 hours]**

Need finding. Data collection (interviews, surveys, observational studies, contextual inquiry). Sampling. Qualitative data analysis (coding, thematic analysis/card sorting, focus group, inter-rater reliability, threads to validity, triangulation). Quantitative data analysis in need finding. Ethical considerations in human studies.

UNIT III **[09 hours]**

Models, theories and frameworks in HCI. What are they and why are they useful? Perception and memory. Model human processor. Information foraging. Activity theory. Distributed cognition. Mental models.

UNIT IV **[09 hours]**

Prototyping (low-fi and hi-fi prototypes, use of metaphors). Evaluation (quantitative – controlled experiments, measures, statistical comparisons; qualitative – heuristics, cognitive walkthroughs, desirability and reaction toolkits, hallway usability study).

UNIT V **[09 hours]**

Recurring problems in HCI and ideas on how to approach them: Human-information interaction (search and browsing, information seeking and sensemaking, minimal learning); Human-AI interaction (role of the AI agent, challenges of non-determinism, explanation, trust, principles, wizard of oz methods);

Suggested Readings:

Human-Computer Interaction – Pearson
Interaction Design: Beyond Human-Computer Interaction – Wiley
The Design of Everyday Things – Basic Books
About Face: The Essentials of Interaction Design – Wiley

References:

Designing the User Interface: Strategies for Effective Human-Computer Interaction – Pearson
Measuring the User Experience – Morgan Kaufmann
Observing the User Experience – Morgan Kaufmann
Human-AI Interaction – Microsoft Research

Course Code : **BTC504** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Computer Graphics**

Course Objectives:

- To understand graphics systems and output primitives
- To learn transformations and projections
- To analyze clipping algorithms

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand graphics fundamentals
- **CO2:** Apply transformations
- **CO3:** Analyze clipping techniques
- **CO4:** Apply curve and surface modeling
- **CO5:** Analyze rendering techniques

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	2	2	2	–	–	–	–	1	–	1
CO5	3	3	3	3	2	–	–	–	–	1	–	2

Course Outline:

UNIT I **[09 hours]**

OVERVIEW OF GRAPHICS SYSTEM:Input devices, Output primitives, Video display devices, Raster and vector graphics, linecircle-ellipse generating algorithm, filled area primitives.

UNIT II **[09 hours]**

TRANSFORMATIONS AND PROJECTIONS: 2D Transformation, 3D transformation, Parallel projection, Perspective projection.

UNIT III **[09 hours]**

LINE CLIPPING: Cohen-Sutherland, Liang Barsky, Polygon clipping: Sutherland Hodgeman &Weiler-Atherton polygon clipping.

UNIT IV **[09 hours]**

CURVES & SURFACES: Conics-Parametric forms for circle, ellipse, parabola; Bezier Curves, Bernstein polynomials, Convex Hull property, B-Spline Curves: Knot vectors uniform and open uniform curves, Periodic B-splines, Uniform B-splines, Non-uniform, rational B-splines, subdividing curves.

UNIT V **[09 hours]**

HIDDEN SURFACE REMOVAL: Hidden Surface Removal: Back face removal, Floating Horizon method for curved objects, Z-Buffer or depth buffer algorithm, Painter's algorithm(Depth sorting method), Binary space partitioning trees, Scan line algorithm, Warnock's algorithm (Area subdivision method).

Suggested Readings:

Numerical Methods for Engineers Author Steven C. Chapra and Raymond P. Canale Publishe r Tata McGraw Hill Edition 1995 2

Introductory methods of Numerical analysis Author S.S.Sastry Publishe r Prentice - Hall of India

Course Code : **BTC505** Course Type : **Humanities** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Managerial Economics and Financial Analysis**

Course Objectives:

- To analyze the fundamental concepts of managerial economics and their role in business decision making.
- To apply demand, supply, elasticity, and forecasting techniques for solving managerial problems.
- To prepare and interpret basic financial accounting statements used in business organizations.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the principles and scope of managerial economics in business decision making.
- **CO2:** Analyze demand, supply, and elasticity concepts for managerial applications.
- **CO3:** Apply demand forecasting methods to estimate future market trends.
- **CO4:** Analyze market equilibrium and pricing decisions under different market conditions.
- **CO5:** Prepare and interpret basic financial accounting statements such as journal, ledger, trial balance, profit and loss account, and balance sheet.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	–	–	–	1	–	–	–	1	1	1
CO2	3	3	1	1	–	–	–	–	–	–	1	1
CO3	2	3	2	2	1	–	–	–	–	1	1	1
CO4	3	3	2	2	–	1	–	–	–	1	1	1
CO5	2	2	1	1	–	–	–	1	–	1	3	1

Course Outline:

UNIT-I **[09 hours]**

Introduction to Managerial Economics: Nature, scope and methods of managerial economics, the economics of a business and managerial decision making, Ten Principles of Economics, circular flow model, production possibility frontier, microeconomics vs macroeconomics.

UNIT-II **[09 hours]**

Demand Analysis: Concept, types and determinants of demand, Law of Demand and its limitations, elasticity of demand – types and measurement, applications of demand and elasticity in managerial decision making.

UNIT-III **[09 hours]**

Supply Analysis and Market Equilibrium: Concept, types and determinants of supply, elasticity of supply, applications of supply and demand, market equilibrium of supply, demand and price, managerial challenges in market equilibrium.

UNIT-IV **[09 hours]**

Demand Estimation and Forecasting: Demand estimation and forecasting, factors governing demand forecasting, methods of demand forecasting – survey methods, expert opinion, Delphi method, controlled experiments, judgemental approach, time series analysis.

UNIT-V **[09 hours]**

Introduction to Financial Accountancy: Introduction to financial accounting, accounting concepts and conventions, journal, ledger, trial balance, preparation of final accounts, profit and loss account and balance sheet.

Suggested Readings:

N. G. Mankiw, *Principles of Economics*, 10th ed. Boston, MA, USA: Cengage Learning, 2024.

N. Ramachandran and R. K. Kakani, *Financial Accounting for Management*, 6th ed. New Delhi, India: McGraw-Hill Education, 2022.

References:

P. G. Keat and P. K. Y. Young, *Managerial Economics: Economic Tools for Today's Decision Makers*, 8th ed. Boston, MA, USA: Pearson Education, 2022.

D. Salvatore, *Managerial Economics: Principles and Worldwide Applications*, 9th ed. New York, NY, USA: Oxford University Press, 2020.

D. N. Dwivedi, *Managerial Economics*, 9th ed. New Delhi, India: PHI Learning, 2021.

Course Code : **BTC506** Course Type : **Core Lab** No. of Credits **: 2.00** No. of Hours : **60**	**Course Title** **Artificial Intelligence and Machine Learning Lab**

Course Objectives:

- To implement and analyze AI algorithms such as search techniques and reasoning methods using programming.
- To apply machine learning algorithms for solving real-world data-driven problems using modern tools.
- To develop and test AI/ML models and evaluate their performance through experiments.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Implement basic AI techniques such as search algorithms and agent-based models.
- **CO2:** Apply knowledge representation and reasoning methods in problem-solving.
- **CO3:** Develop machine learning models for classification, regression, and clustering tasks.
- **CO4:** Use modern tools and libraries (e.g., Python, ML frameworks) for AI/ML applications.
- **CO5:** Analyze and evaluate the performance of AI/ML models using appropriate metrics.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	–	–	1
CO2	3	3	1	2	1	–	–	–	–	–	–	1
CO3	3	3	2	2	2	–	–	–	1	–	–	1
CO4	3	2	1	1	3	–	–	–	–	–	–	1
CO5	3	3	2	2	2	–	–	1	1	1	–	2

List of Experiments:

1. Design and implement a Simple Reflex Agent for a vacuum cleaner environment.
2. Implement Breadth First Search (BFS) algorithm for graph traversal.
3. Implement Depth First Search (DFS) algorithm.
4. Write a program to implement Uniform Cost Search (UCS).
5. Implement *A Search algorithm** for pathfinding problems.
6. Develop a program for Greedy Best First Search.
7. Implement Hill Climbing algorithm.
8. Implement Minimax algorithm with Alpha-Beta pruning.
9. Develop a program for Forward Chaining inference.
10. Implement Backward Chaining algorithm.
11. Build a simple Expert System using rule-based approach.
12. Implement Linear Regression using a dataset.
13. Implement Decision Tree Classifier.
14. Write a program for K-Nearest Neighbors (KNN) classification.
15. Perform K-Means Clustering on a dataset.

Suggested Readings:

Stuart Russell and Peter Norvig, "Artificial Intelligence A Modern Approach", Pearson Education, 2020, Fourth Edition.
Tom M. Mitchell, "Machine Learning", McGraw-Hill Education, 1997, First Edition.

References:

C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
E. Rich and K. Knight, *Artificial Intelligence*, 3rd ed. New York, NY, USA: McGraw-Hill, 2009.
I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

Course Code : **BTC507** Course Type : **Core Lab** No. of Credits **: 2.00** No. of Hours : **60**	**Course Title** **Computer Networks Lab**

Course Objectives:

- To implement and analyze fundamental networking techniques such as framing, error control, and routing algorithms.
- To develop client-server applications using transport layer protocols and socket programming.
- To apply and evaluate application layer protocols and network services.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Implement data link layer techniques such as framing, error detection, and flow control.
- **CO2:** Apply routing algorithms and perform network layer operations such as IP addressing and subnetting.
- **CO3:** Develop client-server applications using TCP and UDP socket programming.
- **CO4:** Implement application layer protocols such as HTTP, FTP, and DNS.
- **CO5:** Analyze and evaluate the performance of network protocols and services

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	–	–	1
CO2	3	3	1	2	2	–	–	–	–	–	–	1
CO3	3	2	2	2	3	–	–	–	1	1	–	1
CO4	3	2	2	1	3	–	–	–	1	1	–	1
CO5	3	3	2	2	2	–	–	1	1	1	–	2

List of Experiments:

1. Study and compare the OSI and TCP/IP reference models by illustrating their layered architecture and identifying the functions of each layer.
2. Implement a program for character stuffing and de-stuffing.
3. Implement a program for bit stuffing and de-stuffing.
4. Write a program to implement Cyclic Redundancy Check (CRC) for error detection.
5. Simulate the Stop-and-Wait protocol for reliable data transfer.
6. Implement the Sliding Window Protocol (Go-Back-N).
7. Write a program to implement Dijkstra's shortest path algorithm.
8. Implement the Distance Vector Routing algorithm.
9. Write a program to perform IP addressing and subnetting calculations.
10. Develop a TCP client-server application using socket programming.
11. Develop a UDP client-server application using socket programming.
12. Write a program to compare the performance of TCP and UDP protocols.
13. Implement a simple HTTP client-server model.
14. Develop a file transfer application (FTP-like) using sockets.
15. Write a program to simulate the DNS lookup process

Suggested Readings:

S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 6th ed. Hoboken, NJ, USA: Pearson, 2021.
J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 8th ed. Boston, MA, USA: Pearson, 2021.

References:

L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2021.
W. Stallings, *Data and Computer Communications*, 10th ed. Upper Saddle River, NJ, USA: Pearson, 2013.
K. R. Fall and W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2011

Course Code : **BTC508** Course Type : **Core Lab** No. of Credits **: 2.00** No. of Hours : **60**	**Course Title** **Compiler Design Lab**

Course Objectives:

- To implement lexical analysis techniques such as token identification, keyword detection, and file-based analysis using programming and tools like LEX.
- To apply parsing techniques and grammar analysis for constructing syntax analyzers and computing FIRST sets.
- To develop compiler components such as symbol tables, code generators, and simple parsing mechanisms.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Implement lexical analysis programs for identifying tokens, keywords, operators, and character frequencies.
- **CO2:** Use tools like LEX to design lexical analyzers and perform text processing tasks.
- **CO3:** Apply grammar concepts to compute FIRST sets and implement parsing algorithms.
- **CO4:** Design and implement symbol tables and basic compiler data structures.
- **CO5:** Develop simple code generation techniques for intermediate representations.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	–	–	1
CO2	3	2	1	1	3	–	–	–	–	–	–	1
CO3	3	3	2	2	1	–	–	–	–	–	–	1
CO4	3	2	2	1	2	–	–	–	1	–	–	1
CO5	3	3	2	2	2	–	–	1	1	1	–	2

List of Experiments:

1. To identify whether a given string is a keyword or not?
2. To count total no. of keywords in a file (taking file from user)
3. To count total no of operators in a file (taking file from user).
4. To count the total occurrence of each character in a given file (taking file from user).
5. Write a C program to insert, delete and display the entries in the Symbol Table.
6. Write a lex program to count blank spaces, words, lines in a given file.
7. Write a C program to find first of any grammar.
8. To implement LEXICAL ANALYZER for IF Statement.
9. To implement parsing algorithms.
10. To implement Code generator.

Suggested Readings:

J. Levine, *Flex & Bison*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2009.
K. C. Louden, *Compiler Construction: Principles and Practice*. Boston, MA, USA: Cengage Learning, 1997.

References:

D. Grune, K. van Reeuwijk, H. E. Bal and C. J. H. Jacobs, *Modern Compiler Design*, 2nd ed. New York, NY, USA: Springer, 2012.

Course Code : **BTC509** Course Type : **SIP** No. of Credits **: 2.00** No. of Hours : **60**	**Course Title** **Summer Internship Programme (SIP)**

Course Objectives:

- To provide students with exposure to a professional work environment and enable them to apply Computer Science and Engineering concepts to real-world problems.
- To develop practical skills in areas such as software development, data analysis, artificial intelligence, networking, cybersecurity, cloud computing, system administration, and related domains.
- To enhance students' professional skills, including documentation, communication, teamwork, problem-solving, and workplace ethics.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Apply Computer Science and Engineering concepts to solve real-world problems in a professional environment.
- **CO2:** Demonstrate practical skills using appropriate tools, technologies, and platforms relevant to the internship domain.
- **CO3:** Analyze workplace practices, project requirements, and the strengths and limitations of theoretical concepts in practical applications.
- **CO4:** Prepare an internship report documenting tasks performed, tools used, skills acquired, observations, and learning outcomes.
- **CO5:** Present and discuss the internship work effectively during seminar/viva-voce evaluation.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	2	2	1	1	1	1	1	2
CO2	3	2	2	2	3	1	1	1	1	1	1	2
CO3	2	3	2	2	2	2	1	2	1	2	1	2
CO4	1	1	1	1	1	1	1	2	1	3	1	2
CO5	1	1	1	1	1	1	1	1	1	3	1	2

Course Outline:

The Summer Internship Programme (SIP) is a 2-credit course offered after the completion of Semester IV. Students who have completed Semester IV shall undergo a minimum of 4 weeks and a maximum of 6 weeks of internship during the semester break before the commencement of Semester V. The internship may be undertaken in industries, research organizations, startups, government organizations, academic institutions, or other approved organizations related to Computer Science and Engineering.

The internship aims to expose students to a professional work environment and enable them to apply classroom concepts to real-world scenarios. It helps students understand the practical relevance, strengths, and limitations of theoretical models and techniques studied during the programme. The internship shall provide hands-on experience in areas such as software development, web and mobile application development, database management, data analysis, artificial intelligence applications, networking, cybersecurity, cloud computing, system administration, and other relevant domains of Computer Science and Engineering.

During the internship, students shall maintain regular documentation of their work, experiences, observations, and findings. The internship report shall include details of the organization, objectives of the internship, tasks assigned, tools and technologies used, work carried out, skills acquired, learning outcomes, challenges faced, and conclusions. The report must be original and shall adhere to the prescribed internship timeline and institutional guidelines.

Each student shall submit four copies of the internship report to the department along with the internship completion certificate issued by the host organization. The report shall include both a preliminary report and a concluding report, wherever applicable. The internship report shall be evaluated for 60 marks.

At the end of the internship, each student shall present the internship work before a panel of evaluators. The presentation and viva-voce shall include an in-depth discussion on the internship work, learning outcomes, technical skills acquired, and relevance of the internship to Computer Science and Engineering. The panel shall assess the student's commitment, understanding, communication skills, and learning from the internship. The presentation and viva-voce shall carry 40 marks.

The total evaluation of the Summer Internship Programme shall be for **100 marks**, consisting of **60 marks for the internship report** and **40 marks for the presentation and viva-voce**. To pass the course, a student must secure a minimum of **50 marks out of 100**, i.e., **50%**, or as prescribed by the University regulation

Course Code : **BTC510** Course Type : **Mandatory** No. of Credits : **0** No. of Hours : **30**	**Course Title** **Constitution of India**

Course Objectives:

- To understand and analyze the fundamental features of the Indian Constitution, including fundamental rights, duties, and directive principles.
- To examine the structure and functioning of Union, State, and Local Governments in India.
- To analyze and evaluate the role of constitutional bodies and democratic institutions in the governance process.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the historical background, basic structure, and salient features of the Indian Constitution.
- **CO2:** Analyze the structure, powers, and functions of the Union Government.
- **CO3:** Explain the organization and functioning of State and Local Governments.
- **CO4:** Analyze the role and responsibilities of local administration and Panchayati Raj institutions.
- **CO5:** Evaluate the role and functioning of the Election Commission and electoral process in India.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	–	–	–	2	–	3	–	1	–	1
CO2	2	3	1	1	–	2	–	2	–	1	1	1
CO3	2	2	1	1	–	3	–	2	1	1	–	1
CO4	2	3	2	2	–	3	1	2	1	1	1	1
CO5	2	3	1	2	–	2	–	3	1	2	1	2

Course Outline:

UNIT-I **[06 hours]**

THE CONSTITUTION - INTRODUCTION: The History of the Making of the Indian Constitution, Preamble and the Basic Structure, and its interpretation, Fundamental Rights and Duties and their interpretation- State Policy Principles.

UNIT-II **[06 hours]**

UNION GOVERNMENT: Structure of the Indian Union-President, Role and Power, Prime Minister and Council of Ministers, Lok Sabha and Rajya Sabha.

UNIT-III **[05 hours]**

STATE GOVERNMENT: Governor, Role and Power, Chief Minister and Council of Ministers, State Secretariat.

UNIT-IV **[06 hours]**

LOCAL ADMINISTRATION: District Administration, Municipal Corporation, Zila Panchayat.

UNIT-V **[07 hours]**

ELECTION COMMISSION: Role and Functioning, Chief Election Commissioner, State Election Commission.

Suggested Readings:

Dr. B. Mahadevan, Chinmaya Vishwa Vidyapeeth, Dr. Vinayak Rajat Bhat, Dr. Nagendra Pavana R.N., Chinmaya Vishwa Vidyapeeth, Dr. Anil Sahasrabudhe, Subhash Kak, Dr. S. Sadagopan, "Introduction to Indian Knowledge System: Concepts and Applications", IIIT Bangalore, 2022.

References:

DD Basu Lexis Nexis, "Introduction to the Constitution of India", 23*rd* Edition, 2018.
B.L. Fadia Sahitya Bhawan, "The Constitution of India", New Edition, 2017.
Rajeev Bhargava, "Ethics and Politics of the Indian Constitution", Oxford University Press, 2008.

SEMESTER-VI

Course Code : **BTC601** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Software Engineering**

Course Objectives:

- To understand and analyze software development life cycle models, requirements engineering, and software design principles.
- To apply software testing, quality assurance, and project management techniques in software development.
- To analyze and evaluate software architectures, usability engineering, and software metrics for developing reliable software systems.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain software engineering principles, process models, and agile development methodologies.
- **CO2:** Analyze and model software requirements using UML and structured modeling techniques.
- **CO3:** Apply software design concepts and architectural patterns for software development.
- **CO4:** Apply software testing, quality assurance, and software metrics for evaluating software quality.
- **CO5:** Analyze software project management, risk management, estimation techniques, and usability engineering principles.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	–	1	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	1	1	–	1
CO3	3	3	3	2	2	–	–	–	1	1	–	1
CO4	3	3	2	3	2	–	–	1	–	1	–	1
CO5	3	3	2	2	1	1	–	1	1	1	2	2

Course Outline:

UNIT-I **[09 hours]**

Introduction to Software Engineering: The evolving role of software, changing nature of software, software developer skills, full stack development and technologies. Process models: Waterfall model, Spiral model, Incremental process models, Evolutionary process models, Agile process models, Incremental development, Test-driven development.

UNIT-II **[09 hours]**

Software Requirements Engineering: Requirement classification, functional and non-functional requirements, user classification and requirements, system requirements, interface specification, software requirements document. Requirements engineering process: feasibility studies, requirements elicitation and analysis, requirements validation, requirements management, documentation – UML. Models: Business process modeling (BPM), Context models, Behavioral models, Data models, Object models, Structured methods.

UNIT-III **[09 hours]**

Software Design Engineering: Design concepts, design process, the design model, pattern-based software design. Creating an architectural design: software architecture, data design – ER diagrams, architectural styles and patterns, architectural design, assessing alternative architectural designs, mapping data flow into software architecture. Modeling component-level design: designing class-based components, conducting component-level design, object constraint language, designing conventional components.

UNIT-IV **[09 hours]**

Software Quality Engineering: Testing strategies, strategic approach to software testing, test strategies for conventional software, Black-box and White-box testing, Validation testing, System testing, Debugging, Object-oriented testing strategies and techniques. Software quality assurance models and metrics. Software metrics: software structure complexity, framework for product metrics, metrics for analysis model, design model, source code, testing, maintenance, process and products.

UNIT-V **[09 hours]**

Software Project Management: Project planning, scheduling, Gantt chart, Critical Path Method (CPM), PERT, Risk management, Estimation models, Quality management, Software quality assurance, Software inspection and reviews, Statistical software quality assurance, Software reliability, ISO quality standards. Usability

Engineering: UI and UX design principles, usability analysis, design steps, usability evaluation methods and techniques.

Suggested Readings:

R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill Education, 2019.

Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson Education, 2016.

References:

Jalote, *Software Engineering: A Precise Approach*. New Delhi, India: Wiley India, 2010.

W. S. Jawadekar, *Software Engineering: A Primer*. New Delhi, India: Tata McGraw-Hill Education, 2008.

R. Mall, *Fundamentals of Software Engineering*, 5th ed. New Delhi, India: PHI Learning, 2018.

D. Jain, *Software Engineering: Principles and Practice*. New Delhi, India: Oxford University Press, 2019.

Y. Wang, *Software Engineering Foundations*. Boca Raton, FL, USA: Auerbach Publications, 2008.

H. van Vliet, *Software Engineering: Principles and Practice*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2008

Course Code : **BTC602** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Data Mining**

Course Objectives:

- To understand and analyze the knowledge discovery process, data preprocessing techniques, and major data mining tasks.
- To apply association rule mining and frequent pattern mining techniques for discovering hidden patterns from large datasets.
- To analyze and evaluate sequential, temporal, graph, and web mining techniques for extracting useful knowledge from complex data.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the knowledge discovery process, data preprocessing techniques, and data mining tasks.
- **CO2:** Apply association rule mining and frequent itemset mining techniques to discover patterns from transactional data.
- **CO3:** Analyze sequential patterns, temporal patterns, and time-series data using suitable mining techniques.
- **CO4:** Apply graph mining techniques to identify frequent subgraphs and clusters in large graph datasets.
- **CO5:** Analyze web mining techniques for mining web content, web structure, and web usage data.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	2	3	3	–	–	–	1	1	–	1
CO5	3	3	2	3	3	1	–	1	1	1	–	2

Course Outline:

UNIT-I **[08 hours]**

Knowledge Discovery and Data Pre-processing: Knowledge Discovery Process, Data Mining Tasks, Data Pre-processing Techniques, Data Cleaning, Data Integration, Data Transformation, Data Reduction, Data Discretization, Data Mining Applications.

UNIT-II **[10 hours]**

Association Rule Mining: Basic concepts of Association Rule Mining, Frequent Itemset Mining, Apriori Algorithm, FP-Growth Algorithm, Mining various kinds of association rules, Constraint-based Association Mining, Correlation Analysis.

UNIT-III **[11 hours]**

Sequential Pattern and Temporal Data Mining: Sequential Pattern Mining concepts, primitives and scalable methods, Transactional Patterns, Temporal-based Frequent Patterns, Mining Time Series Data, Periodicity Analysis for time-related sequence data, Trend Analysis, Similarity Search in Time-Series Analysis.

UNIT-IV **[08 hours]**

Graph Mining: Representation of Graph Data, Mining Frequent Subgraphs, Subgraph Pattern Mining, Graph Clustering, Finding Clusters in Large Graphs, Applications of Graph Mining.

UNIT-V **[08 hours]**

Web Mining: Web Content Mining, Mining Web Page Layout Structure, Mining Web Link Structure, Automatic Classification of Web Documents, Web Usage Mining, Applications of Web Mining.

Suggested Readings:

J. Han, J. Pei and H. Tong, *Data Mining: Concepts and Techniques*, 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2023.

B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, 2nd ed. Berlin, Germany: Springer, 2011.

References:

P.-N. Tan, M. Steinbach, A. Karpatne and V. Kumar, *Introduction to Data Mining*, 2nd ed. Noida, India: Pearson Education, 2018.

C. Chatfield and H. Xing, *The Analysis of Time Series: An Introduction with R*, 8th ed. Boca Raton, FL, USA: Chapman and Hall/CRC, 2025.

G. Dong and J. Pei, *Sequence Data Mining*, 1st ed. New York, NY, USA: Springer, 2007

Course Code : **BTC603** Course Type : **Core Course** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Internet of Things**

Course Objectives:

- To understand and analyze the fundamentals of IoT, M2M communication, IoT architecture, and design principles.
- To examine IoT reference models, architectural views, real-world design constraints, and deployment considerations.
- To analyze and apply IoT communication protocols, service layer protocols, and security mechanisms across different layers.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the fundamentals, design principles, capabilities, and business processes of IoT and M2M systems.
- **CO2:** Analyze IoT reference models, reference architectures, functional views, information views, deployment views, and operational constraints.
- **CO3:** Compare data link layer and network layer protocols used in IoT systems.
- **CO4:** Analyze transport layer, session layer, and application communication protocols used in IoT environments.
- **CO5:** Evaluate service layer protocols and security mechanisms used in IoT architectures

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	1	1	–	–	1	1	1
CO2	3	3	2	2	2	1	1	–	–	1	1	1
CO3	3	3	2	2	2	1	1	–	–	1	–	1
CO4	3	3	2	2	2	1	1	1	–	1	–	1
CO5	3	3	2	3	2	1	1	2	1	1	1	2

Course Outline:

UNIT-I **[09 hours]**

OVERVIEW OF IOT: An Architectural Overview– Building an architecture- Main design principles-needed capabilities- An IoT architecture outline- Standards considerations. M2M and IoT Technology. Fundamentals-Devices and gateways- Local and wide area networking- Data Management-Business processes in IoT-Everything as a Service (XaaS)- M2M and IoT Analytics-Knowledge Management.

UNIT-II **[09 hours]**

REFERENCE MODEL: IoT Architecture-State of the Art – Introduction, State of the art- Reference Model and architecture- IoT reference Model - IoT Reference Architecture- Introduction-Functional View-Information View- Deployment and Operational View- Other Relevant architectural views. Real-World Design Constraints-Introduction-Technical Design constraints-Hardware is popular again-Data representation and visualization, Interaction and remote control.

UNIT-III **[09 hours]**

IoT DATA LINK LAYER & NETWORK LAYER PROTOCOLS: PHY/MAC Layer(3GPP MTC, IEEE 802.11, IEEE 802.15)-Wireless HART- ZWave- Bluetooth Low Energy- Zigbee Smart Energy, DASH7 - Network Layer-IPv4- IPv6- 6LoWPAN- 6TiSCH-ND- DHCP- ICMP- RPL- CORPL- CARP.

UNIT-IV **[09 hours]**

TRANSPORT & SESSION LAYER PROTOCOLS: Transport Layer (TCP- MPTCP- UDP-DCCP-SCTP)-(TLSDTLS)– Session Layer HTTP-CoAP- XMPP-AMQP-MQTT.

UNIT-V **[09 hours]**

SERVICE LAYER PROTOCOLS & SECURITY: Service Layer -OneM2M-ETSI M2M-OMA- BBF – Security in IoT Protocols – MAC 802.15.4 - 6LoWPAN- RPL- Application Layer.

Suggested Readings:

S. Cirani, G. Ferrari, M. Picone and L. Veltri, *Internet of Things: Architectures, Protocols and Standards*, 1st ed. Hoboken, NJ, USA: Wiley, 2018..

B. K. Tripathy and J. Anuradha, Eds., *Internet of Things (IoT): Technologies, Applications, Challenges and Solutions*, 2nd ed. Boca Raton, FL, USA: CRC Press/Routledge, 2027

References:

P. Waher, *Learning Internet of Things*, 1st ed. Birmingham, U.K.: Packt Publishing, 2015.

J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*, 1st ed. Oxford, U.K.: Academic Press, 2014.

D. Uckelmann, M. Harrison and F. Michahelles, Eds., *Architecting the Internet of Things*, 1st ed. Berlin, Germany: Springer, 2011.

Course Code : **BTC604** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Deep Learning**

Course Objectives:

- To learn deep learning architectures and models
- To analyze CNNs, RNNs, and advanced models
- To apply DL in real-world applications

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Understand neural networks
- **CO2:** Apply deep learning models
- **CO3:** Analyze CNN and architectures
- **CO4:** Evaluate advanced DL methods
- **CO5:** Apply DL to real problems

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	3	–	–	–	–	1	–	1
CO3	3	3	2	3	3	–	–	–	–	1	–	1
CO4	3	3	3	3	3	–	–	1	–	1	–	2
CO5	3	3	3	3	3	1	–	1	1	1	–	2

Course Outline:

UNIT I **[09 hours]**

Machine Learning: Fundamentals; Neural Network: Perceptrons, Back Propagation, Over-fitting, Regularization.

UNIT II **[09 hours]**

Deep Networks: Definition, Motivation, Applications; Principal Component Analysis; Restricted Boltzmann Machine; Sparse Auto-encoder; Deep Belief Net; Hidden Markov Model.

UNIT III **[09 hours]**

Convolution Neural Network (CNN): Basic architecture, Activation functions, Pooling, Handling vanishing gradient problem, Dropout, Greedy Layer-wise Pre-training, Weight initialization methods.

UNIT IV **[09 hours]**

Batch Normalization; Different CNN Models: Alex Net, VGG Net, Google Net, Res Net, Dense Net, MIL, Highway Network, Fractal Network, Siamese Net;

UNIT V **[09 hours]**

Graphical Model: Bayes Net, Variational Auto-encoders. Sequence Learning: 1D CNN, Recurrent Neural Network (RNN), Gated RNN, Long short-term memory (LSTM). Generative Modeling: Generative adversarial network. Zero Shot Learning. Applications.

Suggested Readings:

Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016

References:

Michael A. Nielsen, Neural Networks and Deep Learning , Determination Press, 2015.

Yoshua Bengio, Learning Deep Architectures for AI, now Publishers Inc., 2009

Course Code : **BTC604** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Distributed Systems**

Course Objectives:

- To understand concepts and architecture of distributed systems
- To learn communication, synchronization, and coordination techniques
- To design distributed algorithms for real-world problems

Course Outcomes:

After completion of the course, students will be able to

- **CO1**: Explain distributed system concepts and models
- **CO2:** Analyze communication and synchronization techniques
- **CO3:** Apply distributed file systems and naming services
- **CO4:** Evaluate distributed transactions and concurrency
- **CO5:** Design distributed algorithms

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	3	2	–	–	–	–	1	–	1
CO3	3	3	2	2	3	–	–	–	1	1	–	1
CO4	3	3	2	3	3	1	–	1	–	1	–	2
CO5	3	3	3	3	2	–	–	–	1	1	–	2

Course Outline:

UNIT I **[09 hours]**

Basic Concepts: Definition of a distributed systems, Examples, Resource sharing and the Web, Challenges, System models, Architectural and fundamental models, Networking Interprocess communication, External data representation and marshalling, Client-server and Group communication.

UNIT II **[09 hours]**

Distributed Objects and Process: Distributed objects and remote invocation, Communication between distributed objects, Remote procedure call, Events and notifications - The operating system layer, Protection, Processes and Threads, Communication and invocation, OS Architecture. Security techniques, Cryptographic algorithms, Access control, Digital signatures, Cryptography pragmatics, Needham-Schroeder, Kerberos, Securing electronics transaction, IEEE 802.11 WiFi.

UNIT III **[09 hours]**

Operating System Issues Distributed file systems - Name services, Domain name system, Directory and discovery services, Peer to peer systems, Napster file sharing system, Peer to peer middleware routing overlays – Clocks, Events and process states Clock Synchronization - Logical clocks Global states - Distributed debugging - Distributed mutual exclusion - Elections - Multicast communication.

UNIT IV **[09 hours]**

Distributed Transaction Processing Transactions - Nested transactions - Locks - Optimistic concurrency control - Timestamp ordering - Flat and nested distributed transactions - Atomic commit protocols - Concurrency control in distributed transactions - Distributed deadlocks - Transaction recovery - Overview of replication, Distributed shared memory and Web services.

UNIT V **[09 hours]**

Distributed Algorithms Synchronous network model - Algorithms: leader election, maximal independent set - Asynchronous system model: I/O automata, operations on automata, fairness - Asynchronous shared memory model - Mutual exclusion: model, the problem, stronger conditions, lockout-free mutual exclusion algorithms, lower bound on the number of registers - Asynchronous network model - Asynchronous network algorithms: leader election in a ring and an arbitrary network.

Suggested Readings:

1 George Coulouris, Jean Dollimore, and Tim Kindberg, " Distributed Systems Concepts and Design", 5th ed., Pearson Education, 2011.

References:

Andrew S. Tanenbaum, Maartenvan Steen, "Distributed Systems Principles and Paradigms", 2nd ed., Pearson Education, 2006.

Nancy A. Lynch, "Distributed Algorithms", Hardcourt Asia Pvt. Ltd., Morgan Kaufmann, 2000

Course Code : **BTC604** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Embedded Systems**

Course Objectives:

- To introduce the fundamentals of embedded systems, real-time concepts, and ARM architecture.
- To provide knowledge of embedded hardware components including memory, I/O devices, interrupts, and interfacing.
- To enable students to design and implement embedded applications using modern tools and platforms.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the concepts of embedded systems, real-time systems, and ARM architecture.
- **CO2:** Analyze embedded hardware components such as memory, I/O, interrupts, and interfacing techniques.
- **CO3:** Apply embedded programming concepts, RTOS scheduling, and communication mechanisms.
- **CO4:** Illustrate embedded networking, IoT architecture, and communication protocols.
- **CO5:** Design and develop embedded applications using Embedded C, Arduino, and wireless technologies.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	1	2	-	-	-	-	-	-	1
CO3	3	3	3	2	3	-	-	-	1	1	-	2
CO4	2	3	3	2	3	1	1	-	1	1	-	2
CO5	2	3	3	3	3	1	1	1	2	2	1	2

Course Outline:

UNIT I **[09 hours]**

Introduction to Embedded Computing, characteristics of embedded computing applications, concept of real-time systems, challenges in embedded system design, embedded system design process and ARM architecture fundamentals.

UNIT II **[09 hours]**

Embedded computing platforms, bus protocols and bus organization, memory devices and characteristics, memory mapped I/O, I/O devices, timers and counters, watchdog timers, interrupt controllers, interrupt programming and GPIO control.

UNIT III **[09 hours]**

Sensors and actuators, A/D and D/A converters, low power embedded systems, programming embedded systems, processes and threads, scheduling, shared memory communication, message-based communication and real-time operating systems.

UNIT IV **[09 hours]**

Embedded networking fundamentals, distributed embedded architectures, Internet-enabled embedded systems, IoT overview and architecture, interfacing protocols including UART, SPI, I2C and USB.

UNIT V **[09 hours]**

Wireless protocols and applications including Bluetooth, Bluetooth Low Energy, ZigBee and Wi-Fi, wireless sensor networks, programming in Embedded C, Arduino-based embedded applications and case studies in embedded systems.

Suggested Readings:

Alexander G. Dean — *Embedded Systems Fundamentals with ARM Cortex-M Based Microcontrollers*

Wayne Wolf — *Computers as Components: Principles of Embedded Computing System Design*

References:

Lyla B. Das — *Embedded Systems: An Integrated Approach*

Marwedel Peter — *Embedded System Design*

Course Code : **BTC604** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Cyber Security**

Course Objectives:

- To understand security concepts, threats, and vulnerabilities
- To study intrusion detection and prevention methods
- To evaluate security policies and risk management

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain security principles and threats
- **CO2:** Apply cryptographic techniques
- **CO3:** Analyze network and system security
- **CO4:** Evaluate security mechanisms and attacks
- **CO5:** Apply security practices in real scenarios

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	1	–	1	–	1
CO2	3	3	2	2	3	–	–	1	–	1	–	1
CO3	3	3	2	3	3	–	–	1	–	1	–	1
CO4	3	3	2	3	3	1	–	1	–	1	–	2
CO5	3	3	3	3	3	1	–	1	1	1	–	2

Course Outline:

UNIT-I **[09 hours]**

Critical characteristics of Information - NSTISSC Security Model -Components of information System –SDLC – Information assurance - Security Threats and vulnerabilities - Overview of Security threats-– Security Standards.

UNIT-II **[09 hours]**

Classical Cryptography - Symmetric Cryptography- Asymmetric Cryptography - Modern Cryptography – Access Control - DRM – Steganography – Biometrics.

UNIT-III **[09 hours]**

Network Security - Intrusion Prevention, detection and Management - Firewall – Ecommerce Security - Computer Forensics - Security for VPN and Next Generation Networks.

UNIT-IV **[09 hours]**

Host and Application security -Control hijacking, Software architecture and a simple buffer overflow - Common exploitable application bugs, shellcode - Buffer Overflow - Side-channel attacks - Timing attacks, power analysis, cold-boot attacks, defenses – Malware - Viruses and worms, spyware, key loggers, and botnets; defenses auditing, policy - Defending weak applications - Isolation, sandboxing, virtual machines.

UNIT-V **[09 hours]**

Mobile, GSM and Wireless LAN security - Protection measures - Business risk analysis – Information Warfare and Surveillance – Case study on Attack prevention, detection and response.

Suggested Readings:

William Stallings, "Cryptography and Network Security: Principles and Practice", 6th Edition, PHI,2014.

Michael E. Whitman and Herbert J Mattord, "Principles of Information Security", 6th Edition, Vikas Publishing House, 2017.

References:

Bill Nelson, Amelia Phillips, F. Enfinger and Christopher Stuart, "Guide to Computer Forensics and Investigations", 4th Edition, Thomson Course Technology, 2010.

Matt Bishop, "Computer Security: Art and Science", 1st Edition, Addison-Wesley Professional, 2015.

Course Code : **BTC604** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Natural Language Processing**

Course Objectives:

- To understand fundamentals of NLP and text processing
- To learn language models and syntactic analysis
- To apply NLP techniques to real-world problems

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain NLP fundamentals
- **CO2:** Apply language models and tagging
- **CO3:** Analyze syntactic and semantic structures
- **CO4:** Evaluate NLP techniques
- **CO5:** Apply NLP in real applications

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	3	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	2	3	3	–	–	1	–	1	–	2
CO5	3	3	3	3	3	1	–	1	1	1	–	2

Course Outline:

UNIT I **[09 hours]**

Regular Expressions and Automata, N-grams, Basic Text Processing.

UNIT II **[09 hours]**

Part-of-Speech Tagging, Hidden Markov and Maximum Entropy Models, Probabilistic Language Models, Naïve Bayes

UNIT III **[09 hours]**

Formal Grammars of English, Syntactic Parsing, Statistical Parsing, Features and Unification, Language and Complexity~~.~~, Probabilistic Context Free Grammars

UNIT IV **[09 hours]**

The Representation of Meaning, Computational Semantics, Computational Lexical Semantics.

UNIT V **[09 hours]**

Information Extraction, Question Answering and Summarization, Machine Translation, Applications of NLP.

Suggested Readings:

"Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", C. D. Jurafsky and J. Martin.

References:

"Foundations of Statistical Natural Language Processing", Manning and H. Schutze.
"Computational Approaches to Morphology and Syntax (Oxford Surveys in Syntax & Morphology)" 1st Edition, Roark & Sproat.

Course Code : **BTC604** Course Type : **Elective** No. of Credits **: 3.00** No. of Hours : **45**	**Course Title** **Parallel Computing**

Course Objectives:

- To introduce the fundamentals of parallel computing and different forms of parallelism.
- To understand parallel models and algorithms such as PRAM, reduction, and graph algorithms.
- To analyze parallel architectures and performance issues including load balancing and deadlocks.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain concepts of parallel computing, scalability, and types of parallelism.
- **CO2:** Analyze and apply parallel algorithms such as reduction, prefix sums, and graph-based methods.
- **CO3:** Implement parallel programs using OpenMP and MPI with proper data mapping and synchronization.
- **CO4:** Evaluate different parallel architectures and models for efficient algorithm design.
- **CO5:** Design and implement advanced parallel algorithms and explore GPU-based computing solutions.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	2	2	-	-	-	-	-	-	1
CO3	3	3	3	2	3	-	-	-	1	1	-	2
CO4	2	3	3	3	2	-	-	-	1	1	-	2
CO5	2	3	3	3	3	1	1	-	2	2	1	2

Course Outline:

UNIT I **[09 hours]**

Introduction Parallel processing terminology, Pipelining Vs Data parallelism, Control parallelism, Scalability, Control parallel approach, Data parallel approach, Data parallel approach.

UNIT II **[09 hours]**

PRAM model, parallel reduction, prefix sums, list ranking, graph coloring and merging of sorted lists.

UNIT III **[09 hours]**

Programming parallel processes, concepts and applications of parallel programming, OpenMP and MPI basics, mapping data to processors, dynamic load balancing and deadlocks in parallel systems.

UNIT IV **[09 hours]**

Classification of MIMD algorithms, Hypercube SIMD model, 2-D Mesh SIMD model, UMA multiprocessor model, broadcast algorithms and parallel matrix multiplication algorithms.

UNIT V **[09 hours]**

Sorting Enumeration sort, lower bound on parallel sorting, Odd-even transposition sort. Bitonic merge, Quick sort based algorithms, Random read and random write, introduction to GPU computing, applications of parallel computing and recent trends in parallel computing.

Suggested Readings:

Michael Quinn: Parallel Computing-Theory and Practice, MGH.
Ed. Afonso Ferreira and Jose‟ D. P. Rolin, Parallel Algorithms for irregular problems - State of the art, Kluwer Academic Publishers.

References:

Selim G. Akl, The Design and Analysis of Parallel Algorithms, PH International.
Brassard and Bratley, Fundamentals of Algorithms, PHI, New Delhi.

Course Code : **BTC604** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Robotics**

Course Objectives:

- To introduce the fundamentals of robotics, robot components, and configurations.
- To develop understanding of robot control strategies and programming concepts.
- To explore mobile robotics, navigation, and real-world applications.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the fundamentals of robotics, configurations, and kinematic transformations.
- **CO2:** Analyze robot motion, velocities, and trajectory planning techniques.
- **CO3:** Apply knowledge of actuators, sensors, and feedback mechanisms in robotic systems.
- **CO4:** Understand robot control strategies, visual servoing, and programming concepts.
- **CO5:** Design solutions for mobile robotics including localization, mapping, and navigation.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	2	2	-	-	-	-	-	-	1
CO3	3	3	3	2	3	-	-	-	1	1	-	2
CO4	2	3	3	2	3	1	-	-	1	1	-	2
CO5	2	3	3	3	3	1	1	1	2	2	1	2

Course Outline:

UNIT I **[09 hours]**

Introduction to robotics, applications of robotics, robot configurations and components, robot kinematics, position analysis and coordinate transformations.

UNIT II **[09 hours]**

Differential motions and velocities, trajectory planning, robot motion analysis and workspace analysis.

UNIT III **[09 hours]**

Actuators and drives, sensors in robotics, internal and external sensors, simple sensor processing algorithms and sensory feedback mechanisms.

UNIT IV **[09 hours]**

Visual servoing, robot programming concepts, robot control architectures and robot control strategies.

UNIT V **[09 hours]**

Introduction to mobile robotics, localization, mapping, navigation, motion planning and applications of mobile robots.

Suggested Readings:

B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo — *Robotics: Modelling, Planning and Control*.

Course Code : **BTC604** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Game Development Design Thinking**

Course Objectives:

- To understand various idea generation techniques such as brainstorming, mind mapping, and lateral thinking.
- To develop skills in visual communication through drawing and design representation.
- To enable students to apply design principles in creating user-centered and intuitive systems.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Explain the design process, product life cycle, and ethical considerations in design.
- **CO2:** Apply creative thinking techniques to generate and develop innovative design ideas.
- **CO3:** Analyze products using reverse engineering techniques for improved design understanding.
- **CO4:** Illustrate design ideas using drawing fundamentals and visual communication techniques.
- **CO5:** Design user-centered solutions using technical drawing and design thinking principles.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	2	2	-	-	-	-	-	-	1
CO3	3	3	3	2	3	-	-	-	1	1	-	2
CO4	2	3	3	2	3	1	-	-	1	1	-	2
CO5	2	3	3	3	3	1	1	1	2	2	1	2

Course Outline:

UNIT I **[09 hours]**

Design Process: Product Life Cycle, Design Ethics, Design Process - Four Step, Five Step, Twelve Step, Creativity and Innovation in Design Process, Design limitation.

UNIT II **[09 hours]**

Creating and Developing Ideas: Creative Thinking, Generating Design Ideas, Lateral Thinking, Analogies, Brainstorming, Mind mapping, Naminal Group Technique, Synectics, Development of work, Analytical Thinking, Group Activities Recommended.

UNIT III **[09 hours]**

Reverse Engineering: Reverse Engineering for New Understanding about Products, Reasons for Reverse Engineering, Reverse Engineering Process, Step by Step - Case Study.

UNIT IV **[09 hours]**

Drawing to develop design ideas: Many Uses of Drawing, Communication through Drawing, Drawing Basis: Line, Shape/ Form, Value, Colour, Texture.

UNIT V **[09 hours]**

Technical drawing to develop design: Perspective Drawing - One Point Perspective, Two Point Perspective. Isometric Drawing, Orthographic Drawing, Sectional Views.

Suggested Readings:

Everett N McKay, UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication, 2013.

Don Norman, “The Design of Everyday Things: Revised and Expanded Edition”, 2013.

References:

Steve Krug, “Don't Make Me Think: A Common Sense Approach to Web Usability”, 2nd ed., 2005

Jesse James Garrett, “The Elements of User Experience: User-Centered Design for the Web and Beyond", 2nd ed, 2010

Course Code : **BTC605** Course Type : **Humanities** No. of Credits **: 2.00** No. of Hours : **45**	**Course Title** **Soft Skills**

Course Objectives:

- To introduce students to the fundamental concepts and importance of soft skills in academic, professional, and personal scenarios.
- To help students acquire knowledge of various soft skills including communication, teamwork, critical thinking, leadership, adaptability, and emotional intelligence.
- To provide practice and guidance in the application of soft skills for holistic self-development and effective workplace integration.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Identify and define key soft skills relevant to academic, personal, and professional contexts.
- **CO2:** Demonstrate effective verbal, non-verbal, and written communication skills in varied situations.
- **CO3:** Apply teamwork, collaboration, negotiation, and conflict resolution skills in group settings.
- **CO4:** Demonstrate leadership, adaptability, emotional intelligence, and stress management skills for personal and professional development.
- **CO5:** Reflect on and develop interpersonal skills, ethical decision-making, and workplace readiness for effective career growth.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	–	–	–	1	–	1	1	2	–	2
CO2	–	1	–	–	–	1	–	1	2	3	–	2
CO3	–	2	1	–	–	1	–	2	3	2	1	2
CO4	–	2	1	–	–	2	–	2	2	2	1	2
CO5	1	1	–	–	–	2	–	3	2	2	1	3

Course Outline:

UNIT-I **[06 hours]**

Introduction to Soft Skills and Employability: definition and importance of soft skills, role of soft skills employability, career growth and personal development, professional attitude, self-confidence, workplace readiness.

UNIT-II **[06 hours]**

Communication Skills: Verbal and non-verbal communication, listening skills, speaking skills, body language, assertiveness, barriers to effective communication and methods to overcome them.

UNIT-III **[06 hours]**

Presentation and Written Communication Skills: Presentation skills, structure of presentation, audience analysis, use of visual aids, written communication, email writing, report writing and formal correspondence.

UNIT-IV **[06 hours]**

Interpersonal and Teamwork Skills: Building relationships and networking, teamwork, group dynamics, collaboration, conflict resolution, diversity and inclusion in teams, negotiation and persuasion skills.

UNIT-V **[06 hours]**

Leadership, Adaptability and Emotional Intelligence: Leadership styles and qualities, adaptability and flexibility in changing environments, emotional intelligence, self-awareness, self-regulation, empathy, motivation, social skills, stress management and resilience.

Suggested Readings:

J. Butterfield, *Soft Skills for Everyone*, 1st ed. New Delhi, India: Cengage Learning India, 2016.
A. Kaul, *Effective Business Communication*, 2nd ed. New Delhi, India: PHI Learning, 2015.

References:

D. Goleman, *Emotional Intelligence: Why It Can Matter More Than IQ*, 25th Anniversary ed. London, U.K.: Bloomsbury Publishing, 2020.
Harvard Business Review Press, *HBR Guide to Emotional Intelligence*. Boston, MA, USA: Harvard Business Review Press, 2017.
S. E. Lucas and P. Stob, *The Art of Public Speaking*, 2023 release. New York, NY, USA: McGraw Hill, 2023.
S. P. Robbins and T. A. Judge, *Organizational Behavior*, 19th ed. Harlow, U.K.: Pearson Education, 202

Course Code : **BTC606** Course Type : **Core Lab** No. of Credits **: 2.00** No. of Hours : 60	**Course Title** **Software Engineering Lab**

Course Objectives:

- To identify and analyze real-world software problems and document their requirements using Software Requirements Specification (SRS).
- To model and validate software system behavior using use case models and scenario-based testing.
- To apply design patterns for improving software reusability, maintainability, and overall software quality.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Identify and define the scope of a real-world software system to be developed.
- **CO2:** Prepare Software Requirements Specification (SRS) documents for the identified software system.
- **CO3:** Develop use case models and identify functional scenarios for the software system.
- **CO4:** Test the software system based on use case scenarios and validate its functionality.
- **CO5:** Apply suitable design patterns to improve reusability, maintainability, and quality of the software system.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	1	1	1	1	–	–	1	1	1	1
CO2	2	3	2	1	2	–	–	1	1	3	1	1
CO3	3	3	3	2	2	–	–	–	1	2	1	1
CO4	3	3	2	3	2	–	–	1	1	2	1	1
CO5	3	3	3	2	2	–	–	1	1	2	1	2

List of Experiments:

1. Identify a software system that needs to be developed.
2. Document the Software Requirements Specification (SRS) for the identified system.
3. Identify use cases and develop the Use Case model.
4. Test the software system for all the scenarios identified as per the use case diagram.
5. Improve the reusability and maintainability of the software system by applying appropriate design patterns.
6. Implement the modified system and test it for various scenarios.

Suggested Readings:

Roger S. Pressman, "Software Engineering: A Practitioner's Approach", McGraw-Hill, 8*th* Edition, 2019.

Rajib Mall, "Fundamentals of Software Engineering", PHI Learning Pvt. Ltd., 5*th* Edition, 2018.

References:

Ian Sommerville, "Software Engineering", Pearson Publishers, 10*th* Edition, 2016.

Course Code : **BTC607** Course Type : **Core Lab** No. of Credits **: 2.00** No. of Hours : 60	**Course Title** **Data Mining Lab**

Course Objectives:

- To implement data preprocessing, frequent pattern mining, and association rule mining techniques on real-world datasets.
- To apply sequential pattern mining and time-series mining techniques for discovering temporal patterns.
- To analyze graph and web data using graph mining, graph clustering, and web mining techniques.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Apply data preprocessing techniques to prepare datasets for mining tasks.
- **CO2:** Implement frequent pattern mining algorithms such as Apriori and FP-Growth.
- **CO3:** Generate association rules and interpret discovered patterns from transactional datasets.
- **CO4:** Analyze sequential and time-series data using suitable mining techniques.
- **CO5:** Apply graph mining, graph clustering, and web mining techniques to extract useful knowledge from complex data.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	3	–	–	–	–	1	–	1
CO2	3	3	2	2	3	–	–	–	–	1	–	1
CO3	3	3	2	2	3	–	–	–	–	1	–	1
CO4	3	3	2	3	3	–	–	–	–	1	–	1
CO5	3	3	2	3	3	1	–	1	1	1	–	2

List of Experiments:

1. Write a program to perform data preprocessing on a given dataset.
2. Write a program to implement the Apriori algorithm for frequent itemset mining.
3. Write a program to generate association rules from frequent itemsets.
4. Write a program to implement the FP-Growth algorithm for frequent pattern mining.
5. Write a program to implement sequential pattern mining on a sequence dataset.
6. Write a program to perform time-series trend analysis on a given time-series dataset.
7. Write a program to perform similarity search in time-series data.
8. Write a program to perform graph mining on a given graph dataset.
9. Write a program to perform graph clustering on a large graph dataset.
10. Write a program to perform web mining by extracting web content, analyzing web links, or classifying web documents.

Suggested Readings:

J. Han, J. Pei and H. Tong, *Data Mining: Concepts and Techniques*, 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2023.

B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, 2nd ed. Berlin, Germany: Springer, 2011.

References:

P.-N. Tan, M. Steinbach, A. Karpatne and V. Kumar, *Introduction to Data Mining*, 2nd ed. Noida, India: Pearson Education, 2018.

C. Chatfield and H. Xing, *The Analysis of Time Series: An Introduction with R*, 8th ed. Boca Raton, FL, USA: Chapman and Hall/CRC, 2025.

G. Dong and J. Pei, *Sequence Data Mining*, 1st ed. New York, NY, USA: Springer, 2007.

Course Code : **BTC608** Course Type : **Core Lab** No. of Credits **: 2.00** No. of Hours : 60	**Course Title** **Internet of Things Lab**

Course Objectives:

- To understand and implement basic IoT programming using Arduino and Raspberry Pi platforms.
- To interface and communicate with IoT devices using sensors and wireless communication modules such as Zigbee, GSM, and Bluetooth.
- To develop and deploy IoT-based applications by integrating hardware devices, cloud platforms, and data logging mechanisms.

Course Outcomes:

After completion of the course, students will be able to

- **CO1:** Develop basic IoT programs using Arduino and Raspberry Pi platforms.
- **CO2:** Interface Arduino with wireless communication modules such as Zigbee, GSM, and Bluetooth.
- **CO3:** Interface sensors with Raspberry Pi and process sensor data using Python programming.
- **CO4:** Establish communication between Arduino and Raspberry Pi using a wireless medium.
- **CO5:** Configure a cloud platform, upload sensor data, and design a simple IoT-based system.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	3	–	–	–	1	1	–	1
CO2	3	2	2	2	3	1	–	–	1	1	–	1
CO3	3	3	2	2	3	1	1	–	1	1	–	1
CO4	3	3	3	2	3	1	1	–	1	1	–	1
CO5	3	3	3	3	3	2	1	1	2	2	1	2

List of Experiments:

1. Introduction to Arduino platform and programming
2. Interfacing Arduino to Zigbee module
3. Interfacing Arduino to GSM module
4. Interfacing Arduino to Bluetooth Module
5. Introduction to Raspberry PI platform and python programming
6. Interfacing sensors to Raspberry PI
7. Communicate between Arduino and Raspberry PI using any wireless medium
8. Setup a cloud platform to log the data
9. Log Data using Raspberry PI and upload to the cloud platform.
10. Design an IoT based system

Suggested Readings:

S. Cirani, G. Ferrari, M. Picone and L. Veltri, *Internet of Things: Architectures, Protocols and Standards*, 1st ed. Hoboken, NJ, USA: Wiley, 2018..

B. K. Tripathy and J. Anuradha, Eds., *Internet of Things (IoT): Technologies, Applications, Challenges and Solutions*, 2nd ed. Boca Raton, FL, USA: CRC Press/Routledge, 2027.

References:

P. Waher, *Learning Internet of Things*, 1st ed. Birmingham, U.K.: Packt Publishing, 2015.

J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*, 1st ed. Oxford, U.K.: Academic Press, 2014.

D. Uckelmann, M. Harrison and F. Michahelles, Eds., *Architecting the Internet of Things*, 1st ed. Berlin, Germany:

SEMESTER-VII

Course Code : **BTC701** Course Type : **Core** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Cryptography and Network Security**

Course Objectives:

- To introduce the fundamental concepts of cryptography and network security, including security goals and attacks.
- To provide knowledge of modern encryption algorithms, digital signatures, and authentication mechanisms.
- To explore hash functions, message integrity, and cryptographic protocols.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain security goals, cryptographic attacks, and classical encryption techniques.
- **CO2:** Apply symmetric key cryptography concepts including DES, AES, and related mathematical foundations.
- **CO3:** Analyze asymmetric cryptographic techniques such as RSA, Diffie-Hellman, and digital signatures.
- **CO4:** Evaluate message authentication methods, hash functions, and integrity mechanisms.
- **CO5:** Design secure systems using network security protocols and mechanisms like SSL, IPSec, VPN, and firewalls.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	1	-	-	-	1
CO2	3	3	2	1	2	-	-	1	-	-	-	1
CO3	3	3	3	2	2	-	-	1	-	-	-	2
CO4	3	3	2	3	2	-	-	2	-	-	-	2
CO5	3	3	3	3	3	1	1	2	1	1	1	2

Course Outline:

Unit-I: **[09 Hours]**

Introduction Security goals, cryptographic attacks. Mathematics of cryptography: modular arithmetic, Euclidean and extended Euclidean algorithm. Traditional symmetric key ciphers; Monolithic ciphers: addition and multiplication ciphers, Polyalphabetic ciphers: Vigenere’s ciphers, Hill ciphers, playfair ciphers.

Unit-II: **[09 Hours]**

Symmetric key cryptography Mathematics of symmetric key cryptography: Groups, Rings, Fields, GF, Inverse of a number and polynomial using extended Euclidean algorithm. Modern Block ciphers and its components, DES, AES

Unit-III: **[09 Hours]**

Asymmetric key cryptography Mathematics of asymmetric key cryptography: Euler’s Phi-Function, Fermat’s Little Theorem, Euler’s theorem, Chinese remainder theorem. Diffie-Hellman, Digital signature: RSA, Elgamal, Entity authentication

Unit-IV: **[09 Hours]**

Message Integrity and authentication: MAC, HMAC. Cryptographic Hash Function: Merkle-Damgard, MD5, SHA512.

Unit-V: **[09 Hours]**

Network Security Key Management, PGP, IPSec, SSL, Firewalls, Intrusion Detection, Password management, Virus. Virtual Private Network.

Suggested Readings:

Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw-Hill publication, 2nd Edition, 2010.
William Stallings ,“Cryptography and Network Security: Principles and Standards”, Prentice Hall India, 7th Edition, 2017.
John R. Vacca, “Computer and Information Security Handbook”, Morgan Kaufmann Publishers, 3rd Edition, 2017.

References:

Richard H. Baker, Network Security, McGraw Hill International 3 rd Edition,1996.
B. Schneier, Applied Cryptography, John Wiley New York, 2 nd Edition, 1996.
C. Kaufman et. al, Network Security, Prentice Hall International, 2 nd Edition, 2002.

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Image Processing and Computer Vision**

Course Objectives:

- To introduce the fundamentals of image processing and computer vision, including image formation and representation.
- To explore image recognition, object detection, and segmentation techniques.
- To enable students to apply computer vision techniques in real-world applications.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain image formation, filtering techniques, and frequency domain concepts.
- **CO2:** Apply feature detection and matching techniques such as edge detection, SIFT, and Hough transform.
- **CO3:** Analyze camera geometry, stereo vision, and motion estimation techniques.
- **CO4:** Implement methods for image classification, object detection, and segmentation.
- **CO5:** Evaluate and apply computer vision techniques to real-world problems

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	1	–	1
CO2	3	3	2	2	1	–	–	–	–	–	–	1
CO3	3	3	2	2	2	–	–	–	–	–	–	1
CO4	3	2	3	2	2	–	–	–	–	–	–	1
CO5	3	3	2	2	1	–	–	1	1	1	–	2

Course Outline:

Unit-I: **[09 Hours]**

Introduction to Image Processing and Computer Vision, What is an Image and Computer Vision, Image Formation, Camera Projection, Sampling and Aliasing, Image Filtering, Frequency Domain Analysis of Images , Image Pyramids , Wavelets.

Unit-II: **[09 Hours]**

Edge Detection , Feature Points and Corner Detection , Local Image Features, Scale Invariant Feature Transform (SIFT) ,Hough Transform ,Feature Descriptors , Feature Matching.

Unit-III: **[09 Hours]**

Camera Geometry , Camera Calibration — Stereo Vision , Epipolar Geometry , Multiple View Geometry , Image Alignment , RANSAC (Random Sample Consensus).

Unit-IV: **[09 Hours]**

Optical Flow , Motion Estimation Techniques , Motion Analysis , Object Tracking Fundamentals , Feature-based Tracking , Applications of Motion Estimation in Computer Vision.

Unit-V: **[09 Hours]**

Image Classification , Object Detection , Semantic Segmentation , Introduction to Deep Learning-based Vision Applications , Real-world Applications of Computer Vision.

Suggested Readings:

Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd ed., The University of Washington, 2022 Reference: http://szeliski.org/Book/

D. Forsyth and J. Ponce, Computer Vision - A modern approach, 2nd ed, Pearson Education India, 2015 , Reference: http://luthuli.cs.uiuc.edu/~daf/CV2E-site/cv2eindex.html

References:

Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press, 2004 , Reference: http://www.r-5.org/files/books/computers/algo-list/image-processing/vision/Richard_Hartley_Andrew_Zisserman-Multiple_View_Geometry_in_Computer_Vision-EN.pdf

Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016 , Reference: https://www.deeplearningbook.org/

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Deep Computer Vision**

Course Objectives:

- To introduce the fundamentals of image processing and computer vision relevant to deep learning.
- To provide knowledge of feature detection, extraction, and matching techniques.
- To enable students to apply deep learning concepts for image classification, object detection, and segmentation.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain fundamentals of image processing, image formation, and sampling techniques.
- **CO2:** Apply image filtering and frequency domain techniques for image analysis.
- **CO3:** Analyze feature detection and matching methods such as SIFT and Hough transform.
- **CO4:** Evaluate camera geometry, stereo vision, and motion estimation techniques.
- **CO5:** Design and implement deep learning-based solutions for image classification, object detection, and segmentation.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	1	2	-	-	-	-	-	-	1
CO3	3	3	2	2	2	-	-	-	-	-	-	1
CO4	3	3	3	2	2	-	-	-	-	-	-	1
CO5	3	3	3	3	3	1	1	1	1	1	1	2

Course Outline:

Unit-I: **[09 Hours]**

Fundamentals of image processing and computer vision, image formation, camera projection, sampling and aliasing.

Unit-II: **[09 Hours]**

Image filtering, frequency domain analysis of images, pyramids and wavelets.

Unit-III: **[09 Hours]**

Feature detection and matching, edge detection, feature points and corners, local image features, SIFT, Hough transform, feature descriptors and feature matching.

Unit-IV: **[09 Hours]**

Camera geometry and multiple view concepts, camera calibration, stereo vision, epipolar geometry, image alignment, RANSAC and optical flow.

Unit-V: **[09 Hours]**

Recognition and motion estimation, image classification, object detection and semantic segmentation.

Suggested Readings:

Richard Szeliski — *Computer Vision: Algorithms and Applications*

D. Forsyth and J. Ponce — *Computer Vision: A Modern Approach*

References:

Richard Hartley and Andrew Zisserman — *Multiple View Geometry in Computer Vision*

Ian Goodfellow, Yoshua Bengio and Aaron Courville — *Deep Learning*

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Quantum Computing**

Course Objectives:
- To understand quantum computing fundamentals
- To analyze quantum algorithms
- To explore applications of quantum computing

Course Outcomes (COs):

After completion of the course, students will be able to
- **CO1:** Explain quantum computing basics
- **CO2:** Apply quantum gates and circuits
- **CO3:** Analyze quantum algorithms
- **CO4:** Evaluate quantum information processing
- **CO5:** Apply quantum computing concepts

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	2	3	2	–	–	1	–	1	–	2
CO5	3	3	3	3	2	–	–	–	1	1	–	2

Course Outline:

Unit-I: **[09 Hours]**

Fundamentals of Quantum Mechanics for Computing , Limitations of classical computing , Classical vs quantum computation , Mathematical foundations: vectors, matrices, complex numbers, Linear algebra concepts for quantum systems , Postulates of quantum mechanics , Representation of quantum states , Dirac notation , Measurement postulate , Superposition principle

Unit-II: **[09 Hours]**

Qubits and Quantum Gates, Qubit representation, Bloch sphere representation, Multi-qubit systems, Tensor product and composite systems, Quantum gates: Pauli gates (X, Y, Z), Hadamard gate, Phase gate, T gate, Multi-qubit gates: Controlled-NOT (CNOT), SWAP gate, Universal quantum gates, Quantum circuit representation

Unit-III: **[09 Hours]**

Quantum Algorithms, Introduction to quantum algorithms, Quantum parallelism, Deutsch and Deutsch–Jozsa algorithms, Grover's search algorithm, Quantum Fourier Transform, Shor's factoring algorithm (overview), Quantum Phase Estimation,Variational Quantum Eigensolver (VQE), Quantum Approximate Optimization Algorithm (QAOA)

Unit-IV: **[09 Hours]**

Quantum Programming and Simulation, Introduction to quantum programming platforms, Quantum circuit design using Qiskit / Cirq, Quantum simulators, Execution on cloud quantum computers, Implementation of basic quantum algorithms, Noise in quantum systems, NISQ (Noisy Intermediate Scale Quantum) devices

Unit-V: **[09 Hours]**

Applications of Quantum Computing, Quantum cryptography and quantum key distribution, Quantum communication and teleportation, Quantum machine learning concepts, Quantum chemistry simulations, Optimization problems using quantum computing, Emerging quantum technologies and future directions.

Suggested Readings:

Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.
Eleanor Rieffel and Wolfgang Polak, Quantum Computing: A Gentle Introduction, MIT Press.
Jack D. Hidary, Quantum Computing: An Applied Approach, Springer.

References:

N. David Mermin, Quantum Computer Science: An Introduction, Cambridge University Press. Reference Books Phillip Kaye, Raymond Laflamme, and Michele Mosca, An Introduction to Quantum Computing, Oxford University Press. 6. Chris Bernhardt, Quantum Computing for Everyone, MIT Press.

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Virtual Reality**

Course Objectives:

- To introduce the fundamentals of Virtual Reality (VR) and Augmented Reality (AR) systems.
- To understand VR system components, 3D interaction, and user interface design.
- To explore navigation, interaction, and evaluation techniques in virtual environments.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain VR/AR concepts, components, and 3D interaction techniques.
- **CO2:** Analyze VR system design, hardware, and software components.
- **CO3:** Apply modeling techniques and interaction methods in virtual environments.
- **CO4:** Evaluate navigation, usability, and design strategies for 3D user interfaces.
- **CO5:** Design and implement VR/AR applications using open-source tools and frameworks.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	1	2	-	-	-	-	-	-	1
CO3	3	3	3	2	3	-	-	-	1	1	-	2
CO4	2	3	3	3	2	1	1	-	1	1	-	2
CO5	2	3	3	3	3	1	1	1	2	2	1	2

Course Outline:

Unit-I: **[09 Hours]**

Introduction to Virtual Reality, components of a VR system, characteristics of virtual environments, 3D user interfaces, input and output devices and 3D viewing techniques.

Unit-II: **[09 Hours]**

Designing and building VR systems, introduction to Augmented Reality (AR), VR and AR hardware and software components, applications of VR and AR.

Unit-III: **[09 Hours]**

VR modeling techniques including geometric modeling, kinematic modeling, physical modeling and behavior modeling, selection and manipulation during 3D interaction.

Unit-IV: **[09 Hours]**

Travel and wayfinding in virtual environments, strategies for designing and developing 3D user interfaces, evaluation of 3D user interfaces and traditional and emerging VR/AR applications.

Unit-V: **[09 Hours]**

Human factors in virtual reality, case studies on construction of geographic virtual worlds, implementation of virtual and augmented reality applications using open-source toolkits and libraries such as OpenSceneGraph, Vega and VRML.

Suggested Readings:

G.C. Burdea & P. Coiffet, "Virtual reality Technology, Second Ed.", Wiley-India.

GJ Kim, "Designing VR Systems: The Structured Approach", Springer.

References:

D.A. Bowman et al., "3D User Interfaces: Theory and Practice", Addison Wesley.

John Vince, "Virtual Reality Systems", Pearson Ed.

Rick Parent, "Computer Animation: Algorithms & Techniques", Morgan Kaufmann.

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Data Analytics**

Course Objectives:

- To understand statistical foundations for data analytics
- To learn data preprocessing and analytics lifecycle
- To apply analytics using tools like R

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain statistical and analytics concepts
- **CO2:** Apply data preprocessing techniques
- **CO3:** Analyze predictive and descriptive models
- **CO4:** Evaluate big data analytics tools
- **CO5:** Apply analytics using programming tools

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	3	–	–	–	–	1	–	1
CO3	3	3	2	3	3	–	–	–	–	1	–	1
CO4	3	3	2	3	3	–	–	–	1	1	–	2
CO5	3	3	3	3	3	1	–	1	1	1	–	2

Course Outline:

Unit-I: **[09 Hours]**

Mathematics for Data Analytics: Descriptive statistics - Measures of central tendency and dispersion, Association of two variables - Discrete variables, Ordinal and Continuous variable, Probability calculus - probability distributions, Inductive statistics - Point estimation, Interval estimation, Hypothesis Testing - Basic definitions, test

Unit-II: **[09 Hours]**

Introduction to Data Analytics: Introduction to Data Analysis - Analytics, Analytics Process Model, Analytical Model Requirements. Data Analytics Life Cycle overview. Basics of data collection, sampling, preprocessing and dimensionality reduction

Unit-III: **[09 Hours]**

Predictive and Descriptive Analytics: Supervised Learning - Classification, Naive Bayes, KNN, Linear Regression. Unsupervised Learning - Clustering, Hierarchical algorithms – Agglomerative algorithm, Partitional algorithms - K- Means. Association Rule Mining - Apriori algorithm

Unit-IV: **[09 Hours]**

Big Data Analytics: Big Data Overview – State of the practice in analytics, Example Applications - Credit Risk Modeling , Business Process Analytics. Big Data Analytics using Map Reduce and Apache Hadoop, Developing and Executing a Hadoop Map Reduce Program.

Unit-V: **[09 Hours]**

R programming for Data Analysis: Overview of modern data analytic tools.Data Analysis Using R - Introduction to R - R Graphical User Interfaces, Data Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis - Visualization Before Analysis, Dirty Data, Visualizing a Single Variable.

Suggested Readings:

Bart Baesens," Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends”, John Wiley & Sons, 2013.
David Dietrich, “EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data”, John Wiley & Sons, 2015.

References:

Jaiwei Han, MichelineKamber, “Data Mining Concepts and Techniques", Elsevier, 2006.
Christian Heumann and Michael Schomaker, “Introduction to Statistics and DataAnalysis”, Springer, 2016
References 1. Margaret H. Dunham, Data Mining: Introductory and Advanced Topics. Pearson, 2012. 2. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Advanced Compiler Design**

Course Objectives:

- To introduce the advanced concepts of compiler design, architecture, and phases of compilation.
- To understand control flow, data flow analysis, and program analysis techniques.
- To explore compiler optimizations and modern architecture-related issues.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain compiler architecture, phases of compilation, and language processing concepts.
- **CO2:** Analyze control flow and data flow techniques for program optimization.
- **CO3:** Apply intermediate representations such as SSA and symbol table management.
- **CO4:** Evaluate compiler optimization techniques for modern architectures.
- **CO5:** Design advanced compiler components incorporating JIT, garbage collection, and
- parallelization.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	2	2	-	-	-	-	-	-	1
CO3	3	3	3	2	3	-	-	-	1	1	-	2
CO4	3	3	3	3	2	-	-	-	1	1	-	2
CO5	3	3	3	3	3	-	-	-	2	2	1	2

Course Outline:

Unit-I: [09 Hours]

Introduction to compiler construction, compiler architecture, phases of compilation, lexical analysis, syntactic analysis, semantic analysis and overview of modern computer architectures.

Unit-II: [09 Hours]

Control flow and dataflow analysis, basic blocks and loops, dominators, control dependence, reaching definitions, constant propagation and data flow analysis techniques.

Unit-III: [09 Hours]

Symbol table design issues, intermediate representations, high, medium and low level intermediate languages, Static Single Assignment (SSA), Phi functions and applications of SSA in optimization.

Unit-IV: [09 Hours]

Compiler optimizations, scalar optimizations, loop optimizations, dependence analysis, register allocation, code scheduling and optimization techniques for modern architectures.

Unit-V: [09 Hours]

Just-in-time compilation, garbage collection, pointer and alias analysis, code generation, modern compiler frameworks and parallelization concepts.

Suggested Readings:

Advanced Compiler Design and Implementation: Steven Muchnick, 1998.

Compilers: Principles, Techniques, and Tools: A V Aho, Monical Lam, R Sethi, J D Ullman, 2005.

References:

Principles of Program Analysis: Flemming Nielson, Hanne Riis Nielson, Chris Hankin, Springer, 2006.

Modern compiler implementation in Java, Second Edition, Andrew W. Appel, Jens Palsberg, Cambridge University Press, 2002.

The Java programming language, Arnold, Gosling and Holmes, Addison-Wesley, 2005

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Digital Signal Processing**

Course Objectives:

- To understand signals and systems
- To learn transforms and filtering techniques
- To apply DSP in real-world applications

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain signals and systems
- **CO2:** Apply transforms and FFT
- **CO3:** Analyze filtering techniques
- **CO4:** Design digital filters
- **CO5:** Apply DSP in applications

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	3	3	2	–	–	–	–	1	–	2
CO5	3	3	3	3	2	–	–	–	–	1	–	2

Course Outline:

Unit-I: **[09 Hours]**

Basic elements of digital signal Processing: Concept of frequency in continuous time and discrete time signals – Sampling theorem – Discrete time signals. Discrete time systems –Analysis of Linear time invariant systems –Z transform –Convolution and correlation.

Unit-II: **[09 Hours]**

Introduction to DFT: Efficient computation of DFT Properties of DFT – FFT algorithms – Radix-2 and Radix-4 FFT algorithms – Decimation in Time – Decimation in Frequency algorithms – Use of FFT algorithms in Linear Filtering and correlation.

Unit-III: **[09 Hours]**

Structure of IIR: System Design of Discrete time IIR filter from continuous time filter – IIR filter design by Impulse Invariance. Bilinear transformation – Approximation derivatives – Design of IIR filter in the Frequency domain.

Unit-IV: **[09 Hours]**

Symmetric & Anti-symmetric FIR filters: Linear phase filter – Windowing techniques – rectangular, triangular, Blackman and Kaiser windows – Frequency sampling techniques – Structure for FIR systems.

Unit-V: **[09 Hours]**

Finite word length effects in FIR and IIR digital filters: Quantization, round off errors and overflow errors. Multi rate digital signal processing: Concepts, design of practical sampling rate converters, Decimators, interpolators. Polyphase decompositions. Application of DSP – Model of Speech Wave Form – Vocoder.

Suggested Readings:

Oppenheim A V and Sehafer R W, "Discrete Time Signal Processing", Prentice Hall 1989:.
Proakis J G and Manolakis D G, "Digital Signal Processing", Pearson Education India.

References:

Oppenheim A V, Willsky A S and Young I T, "Signal & Systems", Prentice Hall, 1983:. 2. Ifeachor and Jervis, "Digital Signal Processing", Pearson Education India.
DeFatta D J, Lucas J G and Hodgkiss W S, "Digital Signal Processing", J Wiley and Sons, Singapore, 1988 4. Sanjit K Mitra "Digital Signal Processing" TMH

Course Code : **BTC702** Course Type : **Elective** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Pattern Recognition**

Course Objectives:

- To introduce the fundamental concepts and components of pattern recognition systems.
- To understand feature extraction, selection techniques, and distance measures.
- To enable students to apply pattern recognition techniques in real-world domains such as speech and image recognition.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain fundamental concepts, decision functions, and system components of pattern
- recognition.
- **CO2:** Apply feature extraction, selection techniques, and clustering methods.
- **CO3:** Analyze statistical classification methods including Bayes classifier and distance-based approaches.
- **CO4:** Evaluate neural network models and their role in pattern recognition tasks.
- **CO5:** Design and implement pattern recognition solutions for real-world applications.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	1
CO2	3	3	2	2	2	-	-	-	-	-	-	1
CO3	3	3	2	3	2	-	-	-	-	-	-	1
CO4	3	3	3	2	3	-	-	-	1	1	-	2
CO5	3	3	3	3	3	1	1	1	2	2	1	2

Course Outline:

Unit-I: **[09 Hours]**

Fundamental concepts and blocks of a typical pattern recognition system. Decision functions- role and types, pattern and weight space, properties and implementation of decision functions.

Unit-II: **[09 Hours]**

Feature identification, selection and extraction. Distance measures, clustering transformation and feature ordering, clustering in feature selection, feature selection through maximization and approximations.

Unit-III: **[09 Hours]**

Pattern classification by distance functions. Clusters and cluster seeking algorithms. Pattern classification by likelihood functions. Baye's classifier and performance measures.

Unit-IV: **[09 Hours]**

Artificial neural network model, Neural network-based pattern associators, Feed forward networks and training by back-propagation- ART networks.

Unit-V: **[09 Hours]**

Applications of statistical and neural network – based pattern classifiers in speech recognition, image recognition and target recognition.

Suggested Readings:

J.I. Tou & R.C. Gonzalez, Pattern Recognition Priciples, Addition-Wesley.

R. Schalkoff, Pattern Recognition - Statistiucal, Structural and Neural Approaches, John Wiley, 1992.

References:

P.A. Devijer & J. Kittler, Pattern Recognition - A Statistical Approach, Prentice-Hall.

Christopher. M. Bishop, 'Pattern recognition and machine learning, Springer, 2006.

Course Code : **BTC703** Course Type : **Management** No. of Credits : **3.00** No. of Hours : **45**	**Course Title** **Organizational Behaviour**

Course Objectives:

- To introduce the fundamental concepts of Organizational Behaviour and its relevance in modern organizations.
- To understand individual behaviour including personality, perception, motivation, and attitudes.
- To analyze group dynamics, leadership styles, and team effectiveness in organizations.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain the concepts, nature, and scope of Organizational Behaviour.
- **CO2:** Analyze individual behaviour including personality, motivation, and perception.
- **CO3:** Evaluate group dynamics, leadership styles, and team performance.
- **CO4:** Assess organizational culture, structure, and change management processes.
- **CO5:** Apply OB concepts to solve real-world organizational and workplace issues.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	-	-	-	-	-	-	-	2	-	1
CO2	2	2	1	-	-	-	-	-	-	2	-	1
CO3	2	2	2	-	-	-	-	-	2	2	-	1
CO4	2	2	1	1	-	-	-	1	1	2	-	1
CO5	2	2	2	1	-	1	1	1	2	3	1	2

Course Outline:

Unit-I: **[09 Hours]**

Introduction to OB – Definition-Relationship with other fields- Theoretical framework - environmental context-Globalization-Diversity of workforce.

Unit-II: **[09 Hours]**

Foundations of Behaviour: Personality – Meaning-Role of Heredity-Self-Esteem -Self Efficacy -Socialization Process – Person-Situation Interaction-Big 5 Personality Traits -Organization Skills.

Unit-III: **[09 Hours]**

Work Motivation – Meaning-Primary Motive-Secondary Motives – Approaches-Content Theories-Process Theories-Contemporary Theories-Application of Theories in Work Place-Job Design

Unit-IV: **[09 Hours]**

Groups- Nature of Groups- Dynamics of Informal Groups- Functions of Groups and TeamsTeams in Modern Work Place – Leadership – Styles – Strategies - Behavioural.

Unit-V: **[09 Hours]**

Organizational Culture – Definition – Characteristics – Emergence of Cultures -Maintaining A Uniform Culture - Changing Organizational Culture - Culture and Climate Differences - Challenges of Organizations in the Post-Globalized Era.

Suggested Readings:

Fred Luthans (1998) Organizational Behaviour. New Delhi: McGraw-Hill International Edition.
K Aswathappa (2005) Organizational Behaviour Text cases games. New Delhi: Himalaya Publishing Company.
Udai Pareek (2004) Understanding Organizational Behaviour. New Delhi: Oxford University Press.

Course Code : **BTC705** Course Type : **Core Lab** No. of Credits : **2.00** No. of Hours : **60**	**Course Title** **Cryptography and Network Security Lab**

Course Objectives:

- To understand fundamental and classical cryptographic techniques and their mathematical foundations.
- To develop programming skills for implementing symmetric and asymmetric encryption algorithms.
- To apply cryptographic concepts for ensuring confidentiality, integrity, authentication, and secure communication in network systems.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Implement classical cryptographic techniques such as Caesar, Multiplicative, Vigenère, Playfair, and Hill ciphers.
- **CO2:** Apply modular arithmetic concepts including GCD and multiplicative inverse using the Extended Euclidean Algorithm.
- **CO3:** Develop and test symmetric key cryptographic algorithms such as DES and AES for secure data encryption and decryption.
- **CO4:** Implement asymmetric cryptographic algorithms including RSA and Diffie–Hellman key exchange for secure communication.
- **CO5:** Analyze and implement cryptographic hash functions and basic network security mechanisms such as authentication, firewall filtering, and intrusion detection systems.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	3	-	-	-	1	2	-	-
CO2	3	3	2	2	2	-	-	-	1	2	-	-
CO3	3	3	3	2	3	-	-	-	1	2	-	-
CO4	3	3	3	2	3	-	-	-	1	2	-	-
CO5	3	2	3	2	3	-	-	-	2	2	-	-

List of Experiments:

1. Write a program to implement classical encryption techniques such as Caesar Cipher and Multiplicative Cipher.
2. Write a program to implement polyalphabetic ciphers such as Vigenère Cipher and Playfair Cipher.
3. Write a program to implement Hill Cipher using matrix operations for encryption and decryption.
4. Write a program to implement the Extended Euclidean Algorithm to find GCD and multiplicative inverse in modular arithmetic.
5. Write a program to implement a simplified version of Data Encryption Standard (DES).
6. Write a program to perform encryption and decryption using Advanced Encryption Standard (AES).
7. Write a program to implement RSA algorithm for key generation, encryption, and decryption.
8. Write a program to simulate Diffie–Hellman key exchange mechanism.
9. Write a program to implement cryptographic hash functions (MD5 and SHA) for message integrity verification.
10. Write a program to simulate network security mechanisms such as firewall filtering, password authentication, or intrusion detection.

Suggested Readings:

Behrouz A. Forouzan, "Cryptography and Network Security", McGraw-Hill publication, 2nd Edition, 2010.
William Stallings ,"Cryptography and Network Security: Principles and Standards", Prentice Hall India, 7th Edition, 2017.
John R. Vacca, "Computer and Information Security Handbook", Morgan Kaufmann Publishers, 3rd Edition, 2017.

References:

Richard H. Baker, Network Security, McGraw Hill International 3 rd Edition,1996.
B. Schneier, Applied Cryptography, John Wiley New York, 2 nd Edition, 1996.
C. Kaufman et. al, Network Security, Prentice Hall International, 2 nd Edition, 2002.

Course Code : **BTC706** Course Type : **Elective** No. of Credits : **2** No. of Hours : **60**	**Course Title** **Image Processing and Computer Vision LAB**

Course Objectives:

- To understand fundamental concepts of digital image processing and computer vision.
- To develop programming skills for image enhancement, filtering, and transformation techniques.
- To apply feature detection, feature matching, and object recognition methods for real-world visual analysis.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Perform basic image processing operations such as reading, displaying, resizing, and grayscale conversion using standard tools.
- **CO2:** Analyze and demonstrate sampling, quantization, and frequency domain concepts including Fourier Transform and filtering.
- **CO3:** Apply spatial and multi-resolution techniques such as smoothing, sharpening, and image pyramids for image enhancement.
- **CO4:** Implement edge detection, corner detection, and feature extraction techniques for image analysis.
- **CO5:** Apply advanced computer vision techniques such as feature matching, shape detection, and motion estimation for real-world applications.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	–	–	–	–	–	1	–	1
CO2	3	3	2	2	1	–	–	–	–	–	–	1
CO3	3	3	2	2	2	–	–	–	–	–	–	1
CO4	3	2	3	2	2	–	–	–	–	–	–	1
CO5	3	3	2	2	1	–	–	1	1	1	–	2

List of Experiments:

1. Write a program to perform basic image operations such as image reading, display, resizing, and grayscale conversion.
2. Write a program to demonstrate image sampling and quantization, and analyze aliasing effects.
3. Write a program to apply spatial domain filtering techniques such as smoothing and sharpening filters.
4. Write a program to perform frequency domain analysis using Fourier Transform and apply low-pass and high-pass filters.
5. Write a program to construct image pyramids and perform multi-resolution analysis.
6. Write a program to implement edge detection techniques such as Sobel, Prewitt, and Canny.
7. Write a program to detect feature points and corners using Harris Corner Detector.
8. Write a program to extract and match features using SIFT (Scale-Invariant Feature Transform).
9. Write a program to detect shapes using Hough Transform.
10. Write a program to estimate motion using Optical Flow or perform basic object detection/image classification using pre-trained models.

Suggested Readings:

Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd ed., The University of Washington, 2022 Reference: http://szeliski.org/Book/

D. Forsyth and J. Ponce, Computer Vision - A modern approach, 2nd ed, Pearson Education India, 2015 , Reference: http://luthuli.cs.uiuc.edu/~daf/CV2E-site/cv2eindex.html

References:

Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press, 2004 , Reference: http://www.r-5.org/files/books/computers/algo-list/image-processing/vision/Richard_Hartley_Andrew_Zisserman-Multiple_View_Geometry_in_Computer_Vision-EN.pdf

Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016 , Reference: https://www.deeplearningbook.org/

Course Code : **BTC706** Course Type : **Elective** No. of Credits : 2.00 No. of Hours : 60	**Course Title** **Deep Computer Vision Lab**

Course Objectives:

- To understand the principles of image formation, camera geometry, and projection in computer vision.
- To develop practical skills in image processing techniques such as filtering, frequency analysis, and multi-resolution representation.
- To apply feature extraction, feature matching, and deep learning-based vision techniques for solving real-world vision problems.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Perform basic image acquisition, visualization, and understand image formation and representation.
- **CO2:** Analyze camera geometry and simulate projection from 3D world coordinates to 2D image coordinates.
- **CO3:** Apply spatial and frequency domain techniques including filtering, Fourier analysis, and image pyramids.
- **CO4:** Implement feature detection, feature matching, and shape detection techniques for image analysis.
- **CO5:** Apply advanced deep computer vision methods such as optical flow and pre-trained deep learning models for object recognition and motion estimation.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	–	2	–	–	–	1	2	–	-
CO2	3	3	3	2	2	–	–	–	1	2	–	-
CO3	3	3	3	2	3	–	–	–	1	2	–	-
CO4	3	2	3	2	3	–	–	–	1	2	–	-
CO5	3	3	2	3	3	–	–	-	2	2	–	-

List of Experiments:

1. Write a program to perform basic image operations such as image acquisition, grayscale conversion, and visualization of image formation.
2. Write a program to demonstrate camera projection and simulate the transformation from 3D world coordinates to 2D image coordinates.
3. Write a program to illustrate sampling and aliasing effects on images.
4. Write a program to apply spatial domain filtering techniques such as smoothing (Gaussian filter) and sharpening.
5. Write a program to perform frequency domain analysis using Fourier Transform and apply frequency filters.
6. Write a program to construct Gaussian and Laplacian pyramids for multi-resolution image analysis.
7. Write a program to implement edge detection techniques such as Sobel, Prewitt, and Canny.
8. Write a program to detect feature points and corners using Harris Corner Detector and extract local image features (SIFT or ORB).
9. Write a program to perform feature matching between two images and detect shapes using Hough Transform.
10. Write a program to perform motion estimation using Optical Flow or implement object detection/image classification using pre-trained deep learning models

Suggested Readings:

Richard Szeliski — *Computer Vision: Algorithms and Applications*
D. Forsyth and J. Ponce — *Computer Vision: A Modern Approach*
Richard Hartley and Andrew Zisserman — *Multiple View Geometry in Computer Vision*
Ian Goodfellow, Yoshua Bengio and Aaron Courville — *Deep Learning*

Course Code : **BTC706** Course Type : **Elective** No. of Credits : 2.00 No. of Hours : 60	**Course Title** **Quantum Computing Lab**

Course Objectives:

- To understand quantum computing fundamentals
- To analyze quantum algorithms
- To explore applications of quantum computing

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain quantum computing basics
- **CO2:** Apply quantum gates and circuits
- **CO3**: Analyze quantum algorithms
- **CO4:** Evaluate quantum information processing
- **CO5:** Apply quantum computing concepts

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	2	3	2	–	–	1	–	1	–	2
CO5	3	3	3	3	2	–	–	–	1	1	–	2

List of Experiments:

1. Write a program to create a single qubit and display its state vector.
2. Write a program to apply basic quantum gates (X, Y, Z, Hadamard) on a qubit and observe the output.
3. Write a program to visualize a qubit on the Bloch sphere.
4. Write a program to create a two-qubit system and apply a CNOT gate.
5. Write a program to demonstrate quantum superposition using Hadamard gate.
6. Write a program to demonstrate quantum entanglement using Hadamard and CNOT gates.
7. Write a program to design a simple quantum circuit and simulate it using Qiskit.
8. Write a program to implement a basic version of Deutsch algorithm (2-qubit case).
9. Write a program to simulate measurement of qubits and display probabilities.
10. Write a program to run a quantum circuit on a simulator and observe the results (counts/histogram).

Suggested Readings:

Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.
Eleanor Rieffel and Wolfgang Polak, Quantum Computing: A Gentle Introduction, MIT Press.
Jack D. Hidary, Quantum Computing: An Applied Approach, Springer.

References:

N. David Mermin, Quantum Computer Science: An Introduction, Cambridge University Press. Reference Books Phillip Kaye, Raymond Laflamme, and Michele Mosca, An Introduction to Quantum Computing, Oxford University Press. 6. Chris Bernhardt, Quantum Computing for Everyone, MIT Press.
Eric R. Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia, Programming Quantum Computers, O'Reilly.
Scott Aaronson, Quantum Computing Since Democritus, Cambridge University Press.

Course Code : **BTC706** Course Type : **Elective** No. of Credits : 2.00 No. of Hours : 60	**Course Title** **Virtual Reality Lab**

Course Objectives:

- To understand fundamental concepts of virtual reality, 3D graphics, and immersive environments.
- To develop skills in creating and manipulating 3D objects using transformations and viewing techniques.
- To design interactive virtual and augmented reality applications using user input and VR tools.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Create and display basic 3D scenes using primitive shapes and simple rendering techniques.
- **CO2:** Apply 3D geometric transformations and viewing techniques such as translation, rotation, scaling, and projection.
- **CO3:** Develop interactive 3D environments using keyboard and mouse-based user controls.
- **CO4:** Implement object selection, manipulation, and navigation in virtual environments.
- **CO5:** Design and develop basic VR/AR applications using open-source tools and frameworks.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	2	–	–	–	1	2	–	-
CO2	3	3	3	2	2	–	–	–	1	2	–	-
CO3	3	3	3	2	3	–	–	–	1	2	–	-
CO4	3	3	3	2	3	–	–	-	1	2	–	-
CO5	3	3	3	3	2	–	–	–	2	2	–	-

List of Experiments:

1. Write a program to create and display a simple 3D scene using basic shapes (cube, sphere, etc.).
2. Write a program to simulate 3D transformations such as translation, rotation, and scaling.
3. Write a program to demonstrate different 3D viewing techniques (perspective and orthographic projection).
4. Write a program to simulate basic user interaction using keyboard/mouse input in a 3D environment.
5. Write a program to demonstrate a simple Augmented Reality (AR) concept by overlaying text or objects on an image/video.
6. Write a program to design a basic virtual environment using simple VR tools or libraries.
7. Write a program to demonstrate object selection and manipulation in a 3D environment.
8. Write a program to simulate navigation (movement/travel) in a virtual environment using input controls.
9. Write a program to design a simple 3D user interface (menu or object interaction system).
10. Write a program to develop a basic VR/AR application using open-source tools such as OpenSceneGraph or VRML (simple scene description).

Suggested Readings:

G.C. Burdea & P. Coiffet, “Virtual reality Technology, Second Ed.”, Wiley-India.
GJ Kim, “Designing VR Systems: The Structured Approach”, Springer.

References:

D.A. Bowman et al., “3D User Interfaces: Theory and Practice”, Addison Wesley.
John Vince, “Virtual Reality Systems”, Pearson Ed.
Rick Parent, “Computer Animation: Algorithms & Techniques”, Morgan Kaufmann.

Course Code : **BTC706** Course Type : **Elective** No. of Credits : 2.00 No. of Hours : 60	**Course Title** **Data Analytics Lab**

Course Objectives:

- To understand statistical foundations for data analytics
- To learn data preprocessing and analytics lifecycle
- To apply analytics using tools like R

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain statistical and analytics concepts
- **CO2:** Apply data preprocessing techniques
- **CO3:** Analyze predictive and descriptive models
- **CO4:** Evaluate big data analytics tools
- **CO5:** Apply analytics using programming tools

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	3	–	–	–	–	1	–	1
CO3	3	3	2	3	3	–	–	–	–	1	–	1
CO4	3	3	2	3	3	–	–	–	1	1	–	2
CO5	3	3	3	3	3	1	–	1	1	1	–	2

List of Experiments:

1. Write a program to compute measures of central tendency (mean, median, mode) and dispersion (variance, standard deviation) for a dataset.
2. Write a program to analyze the relationship between two variables using correlation and simple visualization.
3. Write a program to perform basic probability distribution analysis (normal/binomial distribution).
4. Write a program to perform data preprocessing tasks such as handling missing values, normalization, and sampling.
5. Write a program to perform dimensionality reduction using a simple technique (feature selection or PCA - basic).
6. Write a program to implement a simple classification algorithm such as K-Nearest Neighbors (KNN).
7. Write a program to implement linear regression for prediction on a dataset.
8. Write a program to perform clustering using K-Means algorithm.
9. Write a program to simulate a simple MapReduce operation (word count or sum of numbers).
10. Write a program in R to perform data analysis including data import, summary statistics, and basic visualization (bar chart, histogram).

Suggested Readings:

Bart Baesens," Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends", John Wiley & Sons, 2013.

David Dietrich, "EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", John Wiley & Sons, 2015.

References:

Jaiwei Han, MichelineKamber, "Data Mining Concepts and Techniques", Elsevier, 2006.

Christian Heumann and Michael Schomaker, "Introduction to Statistics and DataAnalysis", Springer, 2016 References 1. Margaret H. Dunham, Data Mining: Introductory and Advanced Topics. Pearson, 2012. 2. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007

Course Code : **BTC706** Course Type : **Elective** No. of Credits : 2.00 No. of Hours : 60	**Course Title** **Advanced Compiler Design Lab**

Course Objectives:

- To understand the phases of compiler design including lexical, syntax, semantic analysis, and code generation.
- To develop programming skills for implementing compiler construction tools such as parsers, symbol tables, and intermediate code generators.
- To apply optimization and code generation techniques for efficient translation of high-level programs into machine-level code.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Design and implement a lexical analyzer to recognize tokens such as keywords, identifiers, operators, and constants.
- **CO2**: Construct syntax analyzers and perform semantic analysis including type checking for arithmetic expressions.
- **CO3:** Build and analyze control flow graphs and perform basic data flow analysis techniques.
- **CO4:** Design and implement symbol tables and generate intermediate code representations such as three-address code and SSA form.
- **CO5:** Apply code optimization and code generation techniques to improve program efficiency and translate intermediate code into target instructions**.**

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	1	3	–	–	–	1	2	–	-
CO2	3	3	3	2	3	–	–	–	1	2	–	-
CO3	3	3	3	2	3	–	–	–	1	2	–	-
CO4	3	3	3	2	3	–	–	–	1	2	–	-
CO5	3	3	3	3	3	–	–	–	2	2	–	-

List of Experiments:

1. Write a program to implement a lexical analyzer to identify tokens such as keywords, identifiers, operators, and constants.
2. Write a program to implement a simple syntax analyzer (parser) for arithmetic expressions.
3. Write a program to perform basic semantic analysis such as type checking for expressions.
4. Write a program to identify basic blocks and construct a control flow graph (CFG) for a given code snippet.
5. Write a program to perform simple data flow analysis such as reaching definitions or live variable analysis.
6. Write a program to design and implement a symbol table with operations such as insert, delete, and lookup.
7. Write a program to generate intermediate code (three-address code) for simple expressions.
8. Write a program to convert intermediate code into Static Single Assignment (SSA) form (basic version).
9. Write a program to perform simple code optimization techniques such as constant folding and dead code elimination.
10. Write a program to simulate a simple code generation process (convert intermediate code to target-like assembly instructions).

Suggested Readings:

Advanced Compiler Design and Implementation: Steven Muchnick, 1998.
Compilers: Principles, Techniques, and Tools: A V Aho, Monical Lam, R Sethi, J D Ullman, 2005.

References:

Principles of Program Analysis: Flemming Nielson, Hanne Riis Nielson, Chris Hankin, Springer, 2006.
Modern compiler implementation in Java, Second Edition, Andrew W. Appel, Jens Palsberg, Cambridge University Press, 20

Course Code : **BTC706** Course Type : Elective No. of Credits : 2.00 No. of Hours : 60	**Course Title** **Digital Signal Processing Lab**

Course Objectives:

- To understand signals and systems
- To learn transforms and filtering techniques
- To apply DSP in real-world applications

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Explain signals and systems
- **CO2:** Apply transforms and FFT
- **CO3:** Analyze filtering techniques
- **CO4:** Design digital filters
- **CO5:** Apply DSP in applications

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	–	–	–	–	1	–	1
CO2	3	3	2	2	2	–	–	–	–	1	–	1
CO3	3	3	2	3	2	–	–	–	–	1	–	1
CO4	3	3	3	3	2	–	–	–	–	1	–	2
CO5	3	3	3	3	2	–	–	–	–	1	–	2

List of Experiments:

1. Write a program to generate basic discrete-time signals such as unit step, impulse, ramp, and sinusoidal signals.
2. Write a program to perform convolution of two discrete-time signals.
3. Write a program to compute correlation between two signals and analyze similarity.
4. Write a program to compute Discrete Fourier Transform (DFT) of a given signal.
5. Write a program to implement Fast Fourier Transform (FFT) and compare it with DFT.
6. Write a program to demonstrate frequency spectrum analysis using FFT.
7. Write a program to design a simple IIR filter using bilinear transformation method (basic example).
8. Write a program to design an FIR filter using windowing technique (rectangular or Hamming window).
9. Write a program to demonstrate quantization effects (round-off error) on a signal.
10. Write a program to perform basic multirate signal processing such as decimation and interpolation.

Suggested Readings:

Oppenheim A V and Sehafer R W, “Discrete Time Signal Processing”, Prentice Hall 1989:.
Proakis J G and Manolakis D G, “Digital Signal Processing”, Pearson Education India.

References:

Oppenheim A V, Willsky A S and Young I T, “Signal & Systems”, Prentice Hall, 1983:.
Ifeachor and Jervis, “Digital Signal Processing”, Pearson Education India.
3.DeFatta D J, Lucas J G and Hodgkiss W S, “Digital Signal Processing”, J Wiley and Sons, Singapore, 1988 4. anjit K Mitra “Digital Signal Processing” TM

Course Code : **BTC706** Course Type : Elective No. of Credits : 2.**00** No. of Hours : 60	**Course Title** **Pattern Recognition Lab**

Course Objectives:

- To understand the fundamental concepts of pattern recognition, feature extraction, and classification techniques.
- To develop programming skills for implementing clustering, classification, and neural network models.
- To apply machine learning and pattern recognition methods for solving real-world recognition and prediction problems.

Course Outcomes (COs):

After completion of the course, students will be able to

- **CO1:** Implement basic decision functions and visualize pattern spaces with decision boundaries for classification problems.
- **CO2:** Perform feature extraction, normalization, and feature selection for improving pattern recognition performance.
- **CO3:** Apply distance measures and clustering algorithms such as K-Means for grouping and analyzing data patterns.
- **CO4:** Implement classification techniques including Nearest Neighbor and Naive Bayes classifiers.
- **CO5:** Develop simple neural network models and apply pattern recognition techniques to practical applications such as digit recognition and image classification.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	2	–	–	–	1	2	–	-
CO2	3	3	3	2	3	–	–	–	1	2	–	-
CO3	3	3	2	2	3	–	–	–	1	2	–	-
CO4	3	3	3	2	3	–	–	–	1	2	–	-
CO5	3	3	3	3	2	–	–	–	2	2	–	-

List of Experiments:

1. Write a program to implement a simple decision function for classifying data into two classes.
2. Write a program to visualize pattern space and decision boundaries for a small dataset.
3. Write a program to perform feature extraction and normalization on a dataset.
4. Write a program to compute distance measures such as Euclidean and Manhattan distance between feature vectors.
5. Write a program to perform simple feature selection based on variance or correlation.
6. Write a program to implement K-Means clustering algorithm for grouping data.
7. Write a program to classify patterns using distance-based classifier (Nearest Neighbor).
8. Write a program to implement a simple Naive Bayes classifier.
9. Write a program to implement a simple Artificial Neural Network (single-layer or basic multi-layer).
10. Write a program to perform pattern recognition on a simple application such as digit recognition or image classification using a basic model.

Suggested Readings:

J.I. Tou & R.C. Gonzalez, Pattern Recognition Priciples, Addition-Wesley.
R. Schalkoff, Pattern Recognition - Statistiucal, Structural and Neural Approaches, John Wiley, 1992.

References:

P.A. Devijer & J. Kittler, Pattern Recognition - A Statistical Approach, Prentice-Hall.
Christopher. M. Bishop, 'Pattern recognition and machine learning, Springer, 2006

Course Code : **BTC707** Course Type : **Core** No. of Credits **: 4.00**	**Course Title** **Dissertation-I**

Course Objectives:

- To enable students to identify and analyze an emerging research problem or project topic related to Computer Science and Engineering.
- To guide students in conducting a literature survey, identifying research gaps, and formulating objectives, scope, and methodology for the selected problem.
- To develop students' ability to prepare a Phase-I dissertation report and present the proposed work effectively through viva-voce.

Course Outcomes (COs):

By the end of the course, students can be able to;

After completion of the course, students will be able to

- CO1: Identify and define an emerging research problem or project topic in Computer Science and Engineering.
- CO2: Conduct a literature survey and identify research gaps related to the selected problem.
- CO3: Formulate objectives, scope, requirements, and methodology for the proposed dissertation work.
- CO4: Design the proposed system or research framework and prepare an implementation plan.
- CO5: Prepare a Phase-I dissertation report and present the proposed work effectively.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	1	2	1	1	1	1	–	1	1	2
CO2	2	3	1	3	2	1	1	2	–	2	1	3
CO3	3	3	2	2	2	1	1	2	–	2	1	2
CO4	3	3	2	2	2	1	1	1	–	2	2	2
CO5	2	2	1	1	1	1	1	2	–	3	1	2

Course Description

Dissertation-I is the first phase of the final year dissertation work in B.Tech Computer Science and Engineering. Each student shall choose an emerging research problem or project topic related to Computer Science and Engineering and explore it using appropriate research methodologies, tools, and techniques learned during the programme. The student shall work under the guidance of an allotted faculty supervisor.

The work in Dissertation-I shall focus on topic selection, literature survey, problem identification, research gap analysis, objectives formulation, scope definition, requirement analysis, proposed methodology, system design, and preparation of an implementation plan. The student may also carry out preliminary implementation or prototype development, wherever applicable.

At the end of Dissertation-I, the student shall submit a Phase-I dissertation report to the department after obtaining approval from the research supervisor. The report shall include the selected problem, literature review, research gap, objectives, proposed methodology, system design, expected outcomes, work plan, and progress made during the phase.

Assessment shall be carried out by both the internal supervisor and an external expert based on the Phase-I dissertation report and viva-voce presentation. The complete implementation, experimentation, result analysis, final dissertation submission, and final viva-voce shall be carried out in Dissertation-II

Faculty Allotment for the Research Project

Faculty allotment for each student shall be done by the Departmental Internal Advisory Committee based on the student's selected topic or area of interest, the relevance of the topic to Computer Science and Engineering, and the availability of faculty members in the department.

Evaluation of the Research Project

The evaluation of Dissertation-I shall be carried out in two stages for the assignment of marks and grades.

1. Evaluation of Phase-I Dissertation Report

Each student shall submit three copies of the Phase-I dissertation report to the department after obtaining permission from the research supervisor. The report must be checked for plagiarism and should satisfy the plagiarism norms prescribed by the institution.

The evaluation of the Phase-I dissertation report shall carry 60 marks. The average of the marks awarded by the internal supervisor and the external expert shall be considered for the evaluation of the report.

2. Viva-Voce Examination

Each student shall present the Phase-I dissertation work after submission of the report to the department. The viva-voce examination shall be conducted by both the internal supervisor and an external expert.

The viva-voce examination shall carry 40 marks and shall be divided into five categories: (i) Communication skills, (ii) Clarity of subject knowledge, (iii) Presentation skills and body language, (iv) Ability to answer questions, and (v) Overall impression.

Each category shall carry 8 marks. The average of the marks awarded by the internal supervisor and the external expert shall be considered as the viva-voce marks.

Consolidated Evaluation

The department shall consolidate the marks obtained in the Phase-I dissertation report evaluation and viva-voce examination and submit the final marks to the Controller of Examinations.

SEMESTER-VIII

Course Code : **BTC801** Course Type : **Core** No. of Credits **: 8.00**	**Course Title** **Dissertation-II**

Course Objectives:

- To enable students to complete the implementation of the research/project work initiated in Dissertation-I using appropriate tools, technologies, and methodologies.
- To guide students in testing, validating, analyzing, and evaluating the developed system or research solution.
- To develop students' ability to prepare the final dissertation report and present the completed work effectively through viva-voce.

Course Outcomes (COs):

By the end of the course, students can be able to;

After completion of the course, students will be able to

- **CO1:** Implement the proposed system or research solution using suitable tools and technologies.
- **CO2:** Conduct testing, experimentation, and validation of the implemented system or research work.
- **CO3:** Analyze results and evaluate the performance of the developed solution.
- **CO4:** Prepare the final dissertation report following academic and ethical standards.
- **CO5:** Present and defend the completed dissertation work effectively during viva-voce.

COs /POs Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	3	1	1	1	1	1	1	2
CO2	3	3	2	3	3	1	1	1	1	1	1	2
CO3	3	3	2	3	2	1	1	1	1	2	1	2
CO4	2	2	1	2	1	1	1	3	1	3	1	2
CO5	2	2	1	1	1	1	1	2	1	3	1	2

Course Outline:

Dissertation-II is the second and final phase of the dissertation work in B. Tech Computer Science and Engineering. It shall be carried out as a continuation of Dissertation-I. In this phase, each student shall complete the implementation, experimentation, testing, validation, result analysis, and documentation of the research/project work initiated in Dissertation-I under the guidance of the allotted faculty supervisor.

The work in Dissertation-II shall focus on complete system development or research implementation, performance evaluation, testing and validation, analysis of results, comparison with existing methods or systems wherever applicable, and preparation of the final dissertation report. The student shall ensure that the work demonstrates technical relevance, originality, problem-solving ability, and application of Computer Science and Engineering concepts.

At the end of Dissertation-II, each student shall submit the final dissertation report to the department after obtaining approval from the research supervisor. The dissertation report shall include the complete work carried out in both Dissertation-I and Dissertation-II, including introduction, literature review, problem statement, objectives, methodology, system design, implementation, testing, results, discussion, conclusion, limitations, future scope, and references.

Assessment shall be carried out by both the internal supervisor and an external expert based on the final dissertation report and viva-voce presentation.

Faculty Allotment for the Research Project

The faculty supervisor allotted during Dissertation-I shall normally continue to guide the student during Dissertation-II. In exceptional cases, change of supervisor may be permitted by the Departmental Internal Advisory Committee based on valid reasons and availability of faculty members in the department.

Evaluation of the Research Project

The evaluation of Dissertation-II shall be carried out in two stages for the assignment of final marks and grades.

1. Evaluation of Final Dissertation Report

Each student shall submit three copies of the final dissertation report to the department after obtaining permission from the research supervisor. The dissertation report must be checked for plagiarism and should satisfy the plagiarism norms prescribed by the institution.

The evaluation of the final dissertation report shall carry 60 marks. The average of the marks awarded by the internal supervisor and the external expert shall be considered for the evaluation of the dissertation report.

2. Viva-Voce Examination

Each student shall present the completed dissertation work after submission of the final dissertation report to the department. The viva-voce examination shall be conducted by both the internal supervisor and an external expert.

The viva-voce examination shall carry 40 marks and shall be divided into five categories: (i) Communication skills, (ii) Clarity of subject knowledge, (iii) Presentation skills and body language, (iv) Ability to answer questions, and (v) Overall impression.

Each category shall carry 8 marks. The average of the marks awarded by the internal supervisor and the external expert shall be considered as the viva-voce marks.

Consolidated Evaluation

The department shall consolidate the marks obtained in the final dissertation report evaluation and viva-voce examination and submit the final marks to the Controller of Examinations.