Curriculum and Syllabus [Regulation-25]

Incorporating Guidelines of NEP2020

B.Tech. in Computer Science & Engineering

(Effective from 2025-2026 Admission Batch)

JIS College of Engineering

(NAAC 'A' Accredited Autonomous Institute)

(Affiliated to Maulana Abul Kalam Azad University of Technology)

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Curriculum & Syllabus for B. Tech under Autonomy Incorporation of NEP 2020

First Year Curriculum Structure (Effective from 2025-26 admission batch)

Group A: CSE, CSE (ALLIED), IT

Group B: ECE, EE, ECS, CE, ME, AUE, FT, AGR, BME

1st Year 1st Semester (Gr-A)									
Sl. No.	**Broad Category**	**Category**	**Paper Code**	**Subject**	**Contact Hours/ Week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS101	Introduction to Programming and Problem Solving (CSE & Allied)	3	0	0	3	3
2	SCI	Multidisciplinary	PH101	Engineering Physics	3	0	0	3	3
3	SCI	Multidisciplinary	M101	Engineering Mathematics-I	3	0	0	3	3
4	HUM	Value Added Course	HU101	Environmental Science	2	0	0	2	2
5	HUM	Value Added Courses	HU102	Indian Knowledge System	1	0	0	1	1
B. PRACTICAL									
1	ENGG	Major	CS191	Introduction to Programming and Problem-Solving Lab (CSE & Allied)	0	0	3	3	1.5
2	SCI	Skill Enhancement Course	PH191	Engineering Physics Lab	0	0	3	3	1.5
3	ENGG	Skill Enhancement Course	ME194	Engineering Graphics& Computer Aided Design Lab	0	0	3	3	1.5
4	HUM	Ability Enhancement Course	HU191	Communication & Presentation Skill	0	0	3	3	1.5
C. MANDATORY ACTIVITIES / COURSES									
1	Mandatory Course	MC181	Induction Program	0	0	0	0	0	0
Total of Theory, Practical								**24**	**18**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

1st Year 2nd Semester (Gr-A)									
Sl. No.	**Broad Category**	**Category**	**Paper Code**	**Subject**	**Contact Hours/Week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS201	Data structure and Algorithms	3	0	0	3	3
2	ENGG	Minor	CS202	Introduction to Artificial Intelligence	2	0	0	2	2
3	ENGG	Major	CS203	Digital Logic and Computer Organization	3	0	0	3	3
4	SCI	Multidisciplinary	CH201	Engineering Chemistry	2	0	0	2	2
5	SCI	Multidisciplinary	M201	Engineering Mathematics–II	3	0	0	3	3
6	HUM	Value Added Course	HU202	Constitution of India & Professional Ethics	1	0	0	1	1
7	HUM	Ability Enhancement Course	HU203	Design Thinking & Innovation	1	0	0	1	1
B. PRACTICAL									
1	ENGG	Major	CS291	Data structure & Algorithms Lab	0	0	3	3	1.5
2	ENGG	Minor	CS292	Artificial Intelligence Lab	0	0	3	3	1.5
3	ENGG	Major	CS293	Digital Logic and Computer Organization Lab	0	0	3	3	1.5
4	SCI	Skill Enhancement Course	CH291	Engineering Chemistry Lab	0	0	2	2	1
5	ENGG	Skill Enhancement Course	ME293	IDEA LAB Workshop	0	0	3	3	1.5
C. MANDATORY ACTIVITIES / COURSES									
	Mandatory Course	MC281	NSS/ Physical Activities / Meditation & Yoga / Photograph		0	0	0	0	0

			y/ Nature Club						
Total of Theory, Practical								**29**	**22**
Total First Year Credit									**40**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

2nd Year 3rd Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS301	Computer Architecture	3	0	0	3	3
2	ENGG	Major	CS302	Design and Analysis of Algorithms	3	0	0	3	3
3	ENGG	Major	CS303	Operating Systems	3	0	0	3	3
4	ENGG	Major	CS304	Advanced Artificial Intelligence	3	0	0	3	3
5	ENGG	Minor	EC(CS)301	Internet of Things	3	0	0	3	3
6	SCI	Minor	M(CS)301	Discrete Mathematics	3	0	0	3	3
B. PRACTICAL									
1	ENGG	Major	CS391	Computer Architecture Lab	0	0	3	3	1.5
2	ENGG	Major	CS392	Design and Analysis of Algorithms Lab	0	0	3	3	1.5
3	ENGG	Major	CS393	Operating Systems Lab	0	0	3	3	1.5
4	ENGG	Major	CS394	Advanced Artificial Intelligence Lab	0	0	3	3	1.5
5	ENGG	Skill Enhancement Course	CS395	Python Programming Lab	0	1	3	4	2.5
6	ENGG	Minor	EC(CS)391	Internet of Things lab	0	0	3	3	1.5
Total of Theory, Practical								**36**	**28**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

2nd Year 4th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS401	Database Management Systems	3	0	0	3	3
2	ENGG	Major	CS402	Computer Networks	3	0	0	3	3
3	ENGG	Major	CS403	Machine Learning	3	0	0	3	3
4	ENGG	Major	CS404	Formal Language and Automata Theory	3	0	0	3	3
5	SCI	Minor	M(CS)401	Probability and Statistics	3	0	0	3	3
B. PRACTICAL									
1	ENGG	Major	CS491	Database Management Systems Lab	0	0	3	3	1.5
2	ENGG	Major	CS492	Computer Networks Lab	0	0	3	3	1.5
3	ENGG	Major	CS493	Machine Learning Lab	0	0	3	3	1.5
4	ENGG	Minor	M(CS)491	Introduction to R Programming	0	0	3	3	1.5
5	HUM	Ability Enhancement Course	HU(CS)491	Soft Skill & Aptitude	0	0	3	3	1.5
Total of Theory, Practical								**29**	**22.5**
Total Second Year Credit									**50.5**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

3rd Year 5th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS501	Software Engineering	3	0	0	3	3
2	ENGG	Major	CS502	Object Oriented Programming using Java	3	0	0	3	3
3	ENGG	Major	CS503A	Compiler Design	3	0	0	3	3
			CS503B	Cryptography and Network Security					
			CS503C	Computer Graphics					
			CS503D	Data Handling and Visualization					
4	ENGG	Major	CS504	Soft Computing	3	0	0	3	3
5	HUM	Minor	HU(CS)501	Project Management & Finance	2	0	0	2	2
B. PRACTICAL									
1	ENGG	Major	CS591	Software Engineering Lab	0	0	3	3	1.5
2	ENGG	Major	CS592	Object Oriented Programming using Java Lab	0	0	3	3	1.5
3	ENGG	Major	CS593A	Compiler Design Lab	3	0	0	3	1.5
			CS593B	Cryptography and Network Security Lab					
			CS593C	Computer Graphics Lab					
			CS593D	Data Handling and Visualization Lab					
4	ENGG	Major	CS594	Soft Computing Lab	0	0	3	3	1.5
5	PRJ	Project	CS582	Project-I	0	0	4	4	2
Total of Theory, Practical								**30**	**22**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

3rd Year 6th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS601	Web and Internet Technology	3	0	0	3	3
2	ENGG	Major	CS602	Deep Learning	3	0	0	3	3
3	ENGG	Major	CS603A	Image Processing	3	0	0	3	3
			CS603B	Cloud Computing					
			CS603C	Big Data and Data Analytics					
			CS603D	Natural Language Processing					
4	ENGG	Major	CS604A	Mobile Computing	3	0	0	3	3
			CS604B	Human Computer Interaction					
			CS604C	E-Commerce and Digital Business Model					
			CS604D	Quantum Computing					
5	ENGG	Minor	CS605	Cyber Law and Ethics	3	0	0	3	3
B. PRACTICAL									
1	ENGG	Major	CS691	Web and Internet Technology Lab	0	0	3	3	1.5
2	ENGG	Major	CS692	Deep Learning Lab	0	0	3	3	1.5
3	ENGG	Major	CS693A	Image Processing Lab	0	0	3	3	1.5
			CS693B	Cloud Computing Lab					
			CS693C	Data Analytics Lab					
			CS693D	Natural Language Processing Lab					
4	PRJ	Project	CS681	Project-II	0	0	8	8	4
Total of Theory, Practical								**32**	**23.5**
Total Third Year Credit									**45.5**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

4th Year 7th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS701A	Block chain Technology	3	0	0	3	3
			CS701B	Optimization Technique					
			CS701	Bio-informatics					
			CS701D	Robotics					
2	HUM	Minor	HU(CS)701	Human Resource Development and Organizational Behavior	2	0	0	2	2
3	HUM	Value Added Course	HU702	Research Methodology & IPR	1	0	0	1	1
PRACTICAL									
1	PRJ	Project	CS793	Project-III	0	0	12	12	6
2	ENGG	Internship	CS781	Internship (Minimum 1 Month)	0	0	0	0	2
3	ENGG	Skill Enhancement Course	PR792	Rapid Prototyping Lab	0	0	0	4	2
Total of Theory, Practical								**22**	**16**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

4th Year 8th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
B. PRACTICAL									
1	PRJ	PRJ	CS881	Internship/Entrepreneurship	0	0	12	12	6
2	ENGG	Grand Viva	CS882	Grand Viva	0	0	0	0	2
Total of Theory, Practical and Mandatory Activities/Courses								**12**	**8**
Total Fourth Year Credit									**24**

Distribution of credits

Semester	Major	Minor	Multi Disciplinary	Ability Enhancement Course	Skill Enhancement Course	Value Added Course	Project	Internship	Grand Viva	Total
1	4.5	0	6	1.5	3	3	0	0	0	**18**
2	9	3.5	5	1	2.5	1	0	0	0	**22**
3	18	7.5	0	0	2.5	0	0	0	0	**23.5**
4	16.5	4.5	0	1.5	0	0	0	0	0	**25**
5	18	2	0	0	0	0	2	0	0	**23**
6	16.5	3	0	0	0	0	4	0	0	**23.5**
7	3	2	0	0	2	1	6	2	0	**17**
8	0	0	0	0	0	0	6	0	2	**8**
	85.5	**22.5**	**11**	**4**	**10**	**5**	**18**	**2**	**2**	**160**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Curriculum & Syllabus
for
B.Tech under Autonomy (NEP 2020 Implemented)

Computer Science and Engineering

(Effective from 2025-26 admission batch)

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

1st Year 1st Semester (Gr-A)									
Sl. No.	**Broad Category**	**Category**	**Paper Code**	**Subject**	**Contact Hours/ Week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS101	Introduction to Programming and Problem Solving (CSE & Allied)	3	0	0	3	3
2	SCI	Multidisciplinary	PH101	Engineering Physics	3	0	0	3	3
3	SCI	Multidisciplinary	M101	Engineering Mathematics-I	3	0	0	3	3
4	HUM	Value Added Course	HU101	Environmental Science	2	0	0	2	2
5	HUM	Value Added Courses	HU102	Indian Knowledge System	1	0	0	1	1
B. PRACTICAL									
1	ENGG	Major	CS191	Introduction to Programming and Problem-Solving Lab (CSE & Allied)	0	0	3	3	1.5
2	SCI	Skill Enhancement Course	PH191	Engineering Physics Lab	0	0	3	3	1.5
3	ENGG	Skill Enhancement Course	ME194	Engineering Graphics& Computer Aided Design Lab	0	0	3	3	1.5
4	HUM	Ability Enhancement Course	HU191	Communication & Presentation Skill	0	0	3	3	1.5
C. MANDATORY ACTIVITIES / COURSES									
1	Mandatory Course	MC181	Induction Program	0	0	0	0	0	0
Total of Theory, Practical								**24**	**18**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Introduction to Programming and Problem Solving
Course Code: CS101
Contact Hours: 3:0:0
Total Contact Hours: 36
Credits: 3

Course Objectives
By the end of this course, students will be able to:

- Describe the architecture, memory systems, and evolution of computers.
- Convert between number systems and analyze binary arithmetic including IEEE754 representation.
- Construct algorithms and flowcharts for basic computational problems.
- Implement control structures, arrays, pointers, and functions in C programs.
- Demonstrate structured data types and file I/O using the C programming language.

Course Outcomes (COs):
After successful completion of the course the students will be able to
CO1: Describe the architecture, memory hierarchy, and generations of computers, and classify hardware and software components, demonstrating a foundation of engineering knowledge required for understanding computing systems.
CO2: Convert values between number systems and analyze signed and IEEE754 floating-point representations, applying core concepts of mathematics and engineering fundamentals to solve complex engineering problems.
CO3: Construct flowcharts and algorithms for problem solving and develop modular programs in C using appropriate control logic, reflecting skills in design and development of solutions and modern tool usage.
CO4: Implement programs in C using control structures, arrays, pointers, and storage classes, and differentiate between memory management techniques, showcasing proficiency in problem analysis and engineering practice.
CO5: Demonstrate structured data types, file handling, and system-level I/O operations, and evaluate their effectiveness in ensuring data persistence and interfacing with hardware, promoting effective engineering tool usage and lifelong learning.

CO-PO-PSO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	2						2			
CO2	2	2	3	3	3									
CO3	2	3	2	2	2									
CO4	3	2	2	3	3									
CO5	2	2	2	1	1						2			

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content:

Module 1: Basics of Computing & Number Representation (7L)

- History and generations of computers
- Classification: Digital, Analog, Hybrid, Micro, Mini, Mainframe
- Computer architecture: Input/Output units, Memory (Primary & Secondary), CPU
- Number systems: Binary, Octal, Decimal, Hexadecimal
- Conversions among number systems
- Signed number representations: 1's, 2's complement
- Floating point representation: IEEE 754 single & double precision
- ASCII codes
- Overview of compiler, interpreter, assembler

Module 2: Problem Solving & Introduction to C Programming (7 L)

- Algorithm, flowchart, and pseudocode
- Procedural vs Structured programming
- C basics: keywords, identifiers, variable naming (Hungarian Notation)
- Data types, constants, declaration, storage size, endianness
- Operators: Arithmetic, Logical, Relational, Bitwise, Conditional
- Operator precedence and type conversions
- Input/Output: scanf(), printf()

Module 3: Control Structures & Program Design (7 L)

- Control structures: if, if-else, switch, nested conditions
- Loops: while, for, do-while, break, continue
- goto and labels (with discussion on structured vs unstructured programming)
- Functions: declaration, definition, prototypes
- Parameter passing, return types, recursion
- Storage classes: auto, static, extern, register
- Preprocessor directives and macros

Module 4: Arrays, Pointers and Strings (8 L)

- Arrays: 1D & 2D, array to function passing
- Pointers: basics, pointer arithmetic, pointer to arrays
- Strings: character arrays, string library functions, array of strings
- Dynamic memory allocation: malloc(), calloc(), realloc(), free()

Module 5: Structured Data Types, File Handling & System Interface (7 L)

- Structures: definition, initialization, array of structures, pointers to structures

- Unions and enum, typedef, bit fields
- File I/O in C: fopen(), fclose(), fprintf(), fscanf(), fgetc(), fputc()
- Command line arguments

Textbook:

1. Schaum's Outline of Programming with C by Byron S. Gottfried, McGraw-Hill Education, 1st Edition (1996)
2. Let Us C by Yashavant Kanetkar, BPB Publications, 17thEdition
3. Computer Fundamentals by P.K. Sinha and Priti Sinha, BPB Publications,6th Edition

Reference Books:

1. The C Programming Language by Brian W. Kernighan and Dennis M. Ritchiem, Prentice Hall,2ndEdition
2. Fundamentals of Computers by V. Rajaraman and Neeharika Adabala, PHI Learning, 6th Edition
3. Computer Organization and Architecture: Designing for Performance by William Stallings, Pearson Education,10th Edition
4. Mastering C by K. R. Venugopal and S. R. Prasad, Tata McGraw-Hill Education, 2ndEdition
5. Programming in ANSI C by E. Balagurusamy, McGraw Hill Education 8thEdition

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Engineering Physics
Course Code: PH101
Contact: (3:0:0)
Total Contact Hours: 36
Credits: 3
Prerequisites: Knowledge of Physics up to 12th standard.

Course Objectives:
The aim of courses in Physics-I is to provide adequate exposure and develop insight about the basic principles of physical sciences and its practical aspects which would help engineers to learn underlying principles of various tools and techniques they use in core engineering and related industrial applications. The course would also inculcate innovative mindsets of the students and can create awareness of the vital role played by science and engineering in the development of new technologies.

Course Outcome(s):
CO1: Explain the principles of lasers, fibre optics, and holography and apply them in modern optical and communication systems.
CO2: Identify different crystal structures and compute structural parameters such as Miller indices and packing factors; distinguish between metals, semiconductors, and insulators using band theory.
CO3: Utilize the principles of quantum theory—including quantization, wave-particle duality, and Schrödinger equation—to interpret fundamental quantum phenomena.
CO4: Illustrate the basic concepts of statistical mechanics and examine their implications on microscopic particle behaviour.
CO5: Describe the properties of nanomaterials and display/storage devices and analyze their applications in modern technology.

CO-PO Mapping:

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

Course Content:
Module 1 (11L)
Modern Optics
1.01- Laser(6L): Concepts of various emission and absorption processes, Einstein A and B coefficients and
equations, working principle of laser, metastable state, population inversion, condition

necessary for active laser action, optical resonator, illustrations of Ruby laser, He-Ne laser, Semiconductor laser, applications of laser, related numerical problems.

1.02-Fibre Optics(3L): Principle and propagation of light in optical fibers (Step index, Graded index, single and multiple modes) - Numerical aperture and Acceptance angle, Basic concept of losses in optical fiber, related numerical problems.

1.03-Holography(2L): Theory of holography (qualitative analysis), viewing of holography, applications

Module 2 (5L)
Solid State Physics
2.01 Crystal Structure(3L): Structure of solids, amorphous and crystalline solids (definition and examples), lattice, basis, unit cell, Fundamental types of lattices –Bravais lattice, simple cubic, fcc and bcc lattices, Miller indices and miller planes, co-ordination number and atomic packing factor, Bragg's equation, applications, numerical problems.

2.02 Semiconductor(2L): Physics of semiconductors, electrons and holes, metal, insulator and semiconductor, intrinsic and extrinsic semiconductor, p-n junction.

Module 3 (14L)
Quantum and Statistical Mechanics
3.01 Quantum Theory(**5L**): Inadequacy of classical physics-concept of quantization of energy, particle concept of electromagnetic wave (example: Black body radiation, Photoelectric and Compton Effect: no derivation required), wave particle duality; phase velocity and group velocity; de Broglie hypothesis; Davisson and Germer experiment, related numerical problems.
3.02 Quantum Mechanics 1(**4L**)**:** Concept of wave function, physical significance of wave function, probability interpretation; normalization of wave functions-Qualitative discussion; uncertainty principle, relevant numerical problems, Introduction of Schrödinger wave equation (only statement).

3.03Statistical Mechanics (5L):
Concept of energy levels and energy states, phase space, microstates, macrostates and thermodynamic probability, MB, BE, FD, statistics (Qualitative discussions)-physical significance, conception of bosons, fermions, classical limits of quantum statistics, Fermi distribution at zero & non-zero temperature, Concept of Fermi level-Qualitative discussion.

Module 4 (4L)
Physics of Nanomaterials
Reduction of dimensionality, properties of nanomaterials, Quantum wells (two dimensional), Quantum wires (one dimensional), Quantum dots (zero dimensional); Quantum size effect and Quantum confinement. Carbon allotropes. Application of nanomaterials (CNT, graphene, electronic, environment, medical).

Module 5 (2L)
Storage and display devices
Different storage and display devices-Magnetic storage materials, Operation and application of CRT, CRO, LED and OLED.

Text books:
1. Concepts of Modern Engineering Physics- A. S. Vasudeva. (S. Chand Publishers)
2. Engineering Physics - Rakesh Dogra
3. Introduction to Nanoscience and Nanotechnology, An Indian Adaptation-Charles P. Poole, Jr., Frank J. Owens.

Reference books:
1. Optics - Ajay Ghatak (TMH)
2. Solid state Physics - S. O. Pillai
3. Quantum mechanics -A.K. Ghatak and S Lokenathan
4. Fundamental of Statistical Mechanics: B. B. Laud
6. Perspective & Concept of Modern Physics—Arthur Beiser

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Engineering Mathematics-I
Course Code: M 101
Contact (L: T: P): 3 : 0 : 0
Total Contact Hours: 36
Credit: 3

Prerequisites: The students to whom this course will be offered must have the understanding of (10+2) standard algebraic operations, coordinate geometry, and elementary calculus concepts including limits, continuity, differentiation, and integration.

Course Objectives: The objective of this course is to familiarize the prospective engineers with techniques in matrix algebra and calculus. It aims to equip the students with standard concepts and tools at an intermediate to advanced level that will serve them well towards tackling more advanced level of mathematics and applications that they would find useful in their disciplines.

Course Outcomes (COs):
On successful completion of the learning sessions of the course, the learner will be able to:

CO1. Apply linear algebra methods to perform matrix operations, classify matrix structures, solve systems of linear equations, and compute eigenvalues and eigenvectors in engineering contexts.
CO2. Apply differential and integral calculus to evaluate and approximate the behavior of single- variable and multivariable real-valued functions relevant to engineering scenarios.
CO3. Analyze the properties of eigen values and eigen vectors to assess matrix diagonalizability and interpret linear transformations using the Cayley-Hamilton theorem in engineering systems.
CO4. Analyze single-variable and multivariable real-valued functions using differential and integral calculus to model and interpret complex behavior in engineering applications.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	-	-	-	-	-	-	-	-	1
CO2	3	2	-	-	-	-	-	-	-	-	1
CO3	3	3	1	1	-	-	-	-	-	-	2
CO4	3	3	1	1	-	-	-	-	-	-	2
M 101	**3**	**2.5**	**1**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**1.5**

Weightage Values: Strongly mapped: ‘3’, Moderately mapped: ‘2’, Weakly mapped: ‘1’, Not mapped: ’-‘.

Course Content:
Module I: Liner Algebra (11L)

Echelon form and normal (canonical) form of a matrix; Inverse and rank of a matrix; Consistency and inconsistency of system of linear equations, Solution of system of linear equations; Eigenvalues and eigenvectors; Diagonalization of matrix, Cayley-Hamilton theorem.

Module II: Single Variable Calculus (5L)
Rolle's Theorem, Mean value theorems, Taylor's and Maclaurin theorems with remainders; Taylor's series.

Module III: Multivariable Calculus (Differentiation) (13L)
Function of several variables; Concept of limit, continuity and differentiability; Partial derivatives, Total derivative and its application; chain rules, Derivatives of implicit functions Euler's theorem on homogeneous function; Jacobian; Maxima and minima of functions of two variables.

Module IV: Multivariable Calculus (Integration) (7L)
Double Integral, Triple Integral; Change of order in multiple integrals; Line Integral, Surface Integral, Volume Integral. Change of variables in multiple integrals.

Text Books:
1. Grewal, B.S., Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.
2. Kreyszig, E., Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.

Reference Books:
1. Guruprasad, S. A text book of Engineering Mathematics-I, New age International Publishers.
2. Ramana, B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
3. Veerarajan, T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
4. 4. Bali, N.P. and Goyal, M., A text book of Engineering Mathematics, Laxmi Publications,
   Reprint, 2008.
5. Thomas, G.B. and Finney, R.L., Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
6. Apostol, M., Calculus, Volumes 1 and 2 (2nd Edition), Wiley Eastern, 1980.
7. Kumaresan, S., Linear Algebra - A Geometric approach, Prentice Hall of India, 2000.
8. Poole, D., Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005.
9. Bronson, R., Schaum's Outline of Matrix Operations. 1988.
10. Piskunov, N., Differential and Integral Calculus, Vol. I & Vol. II, Mir Publishers, 1969.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Environmental Science
Course Code: HU 101
Credits: 2
Contact Hours: 24
Prerequisites: 10+2

Course Objective (s)

This course will enable the students to,

- Realize the importance of environment and its resources.
- Apply the fundamental knowledge of science and engineering to assess environmental and health risk.
- Know about environmental laws and regulations to develop guidelines and procedures for health and safety issues.
- Solve scientific problem-solving related to air, water, land and noise pollution.

Course Outcome

CO1: Able to understand the natural environment and its relationships with human activities
CO2: The ability to apply the fundamental knowledge of science and engineering to assess environmental and health risk
CO3: Ability to understand environmental laws and regulations to develop guidelines and procedures for health and safety issues
CO4: Acquire skills for scientific problem-solving related to air, water, noise & land pollution.

Module 1 - Resources and Ecosystem (6L)

1.1 Resources (4L)

Types of resources, Human resource, Population Growth models: Exponential Growth, Logistic growth curve with explanation. Maximum Sustainable Yield [Derivation]
Alternative sources of Energy [Solar energy, tidal energy, geothermal energy, biomass energy]

1.2 Ecosystem (2L)

Components of ecosystem, types of ecosystem, Forest ecosystem, Grassland ecosystem, Desert ecosystem, Pond eco system, Food chain, Food web.

Module 2 – Environmental Degradation (10L)

2.1 Air Pollution and its impact on Environment (3L)

Air Pollutants, primary & secondary pollutants, Criteria pollutants, Smog, Photochemical smog and London smog, Greenhouse effect, Global Warming, Acid rain, Ozone Layer Depletion.

2.2 Water Pollution and its impact on Environment (4L)

Water Pollutants, Oxygen demanding wastes, heavy metals, BOD [Rate equation], COD, Eutrophication, Hardness, Alkalinity, TDS and Chloride, Heavy metal (As, Hg, Pb)

poisoning and toxicity. Numerical on BOD, Hardness.

2.3 Land Pollution and its impact on Environment (1L)

Solid wastes, types of Solid Waste, Municipal Solid wastes, hazardous wastes, bio-medical wastes, E-wastes,

2.4 Noise Pollution and its impact on Environment (2L)

Types of noise, Noise frequency, Noise pressure, Measurement of noise level and decibel (dB) Noise intensity, Noise Threshold limit, Effect of noise pollution on human health. Numerical on Measurement of noise level and decibel (dB) and Noise Threshold limit.

Module 3 – Environmental Management (6L)

3.1 Environmental Impact Assessment (1L)

Environmental Auditing, Environmental laws and Protection Acts of India, carbon footprint, Green building practices. *(GRIHA norms)*

3.2Pollution Control and Treatment (2L)

Air Pollution controlling devices, Catalytic Converter, Electrostatic Precipitator.

Waste Water Treatment (Surface water treatment &Activated sludge process), Removal of hardness of water (Temporary &Permanent -Permutit process).

3.3Waste Management (3L)

Solid waste management, Open dumping, Land filling, incineration, composting & Vermicomposting, E-waste management, and Biomedical Waste management.

Module 4 – Disaster Management (2L)

4.1 Study of some important disasters (1L)

Natural and Man-made disasters, earthquakes, floods drought, landslide, cyclones, volcanic eruptions, tsunami, oil spills, forest fires.

4.2Disaster Management Techniques (1L)

Basic principles of disaster management, Disaster Management cycle, Disaster management policy, Awareness generation program

Text Books:

1. Basic Environmental Engineering and Elementary Biology (For MAKAUT), Gourkrishna Dasmohapatra, Vikas Publishing.
2. Basic Environmental Engineering and Elementary Biology, Dr. Monindra Nath Patra &Rahul Kumar Singha, Aryan Publishing House.
3. Textbook of Environmental Studies for Undergraduate Courses, Erach Barucha for UGC, Universities Press

Reference Books:

1. A Text Book of Environmental Studies, Dr. D.K. Asthana & Dr. Meera Asthana, S.Chand Publications.
2. Environmental Science (As per NEP 2020), Subrat Roy, Khanna Publisher

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Indian Knowledge System
Course Code: HU102
Contact: 1:0:0
Credit: 01
No. of lectures: 12

Course outcome: On completing this course the student will be able

CO1: To define, identify, describe and classify the philosophical, literary and socio-religious heritage of ancient India and the core concepts of the Vedic corpus and way of life.

CO2: To discover, enumerate, compare, contrast and categorize the importance of pioneering developments in science and mathematics and evaluate their continuing relevance.

CO3: To analyze, appraise, correlate and describe the ancient Indian heritage in science and technology and examine technological correlations with present-day technological applications.

CO4: To discover, assess and describe traditional knowledge in health care, architecture, agriculture and other sectors and to explore the history of traditional Indian art forms.

CO-PO Mapping:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	-	-	-	-	-	1	2	-	3	-	3
CO2	2	-	1	-	-	2	-	-	3	-	3
CO3	3	1	1	1	2	3	-	-	3	-	3
CO4	2	-	1	-	2	3	-	-	3	-	3

Module-1 (3L)
An overview of Indian Knowledge System (IKS): Importance of Ancient Knowledge - Definition of IKS - Classification framework of IKS - Unique aspects of IKS.
The Vedic corpus: Vedas and Vedangas - Distinctive features of Vedic life.
Indian philosophical systems: Different schools of philosophy (Orthodox and Unorthodox).

Module-2 (3L)
Salient features of the Indian numeral system: Developments in Indian Mathematics in ancient India - Importance of decimal representation - The discovery of zero and its importance - Unique approaches to represent numbers- Contribution of ancient Indian mathematicians

Highlights of Indian Astronomy: Historical development of astronomy in India- key contributions of ancient Indian astronomers.

Module-3 (3L)
Indian science and technology heritage: Metals and metalworking - Mining and ore extraction –Structural engineering and architecture in ancient India: planning, materials, construction and approaches- Dyes and painting; Shipbuilding.

Module-4 (3L)
Traditional Knowledge in Different Sectors: Traditional knowledge and engineering. Traditional Agricultural practices (resources, methods, technical aids); Traditional Medicine and Surgery; History of traditional Art forms and Culture.

Text Books:
1. Amit Jha. *Traditional Knowledge System in India*. New Delhi: Atlantic Publishers, 2024.
2. B. Mahadevan, Vinayak Rajat Bhat, Nagendra Pavana. *Introduction to Indian Knowledge System: Concepts and Applications*. New Delhi: PHI, 2022.
3. Angad Godbole. *Science and Technology in Ancient India*. New Delhi: Biblia Implex, 2023.
4. Pritilakshmi Swain. *Indian Knowledge System*. New Delhi: Redshine Publication, 2024.
5. Vishnudut Purohit. *Fundamentals of Indian Knowledge System*. New Delhi: ABD Publishers, 2024.

Reference Books:
1. L. Basham. *The Wonder that was India*. Vol. I. New Delhi: Picador, 2019.
2. Arun Kumar Jha and Seema Sahay ed. *Aspects of Science and Technology in Ancient India.* Oxford and New Delhi: Taylor and Francis, 2023.
3. Kapil Kapoor and Awadhesh Kumar Singh. *Indian Knowledge Systems*. Vols. 1 and 2. New Delhi: D. K. Printworld, 2005.
4. S. N. Sen and K. S. Shukla, *History of Astronomy in India*. New Delhi: Indian National Science Academy, 2nd edition, 2000.
5. Arpit Srivastava. *Indian Knowledge System*. Rewa: AKS University, 2024.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Introduction to Programming and Problem Solving Lab
Course Code: CS191
Contact Hours: 0:0:3
Total Contact Hours: 36
Credits: 1.5

Course Objectives

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

- Understand the fundamentals of programming logic through algorithmic thinking.
- Implement and debug C programs using various control structures.
- Apply memory management concepts using pointers and arrays.
- Develop structured programs involving functions and recursion.
- Demonstrate file operations and manipulate data using structures and pointers.

Course Outcomes (COs)

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

Course Outcome(s):
CO1: Identify fundamental programming constructs such as data types, operators, control structures, and apply them to solve basic computational problems.
CO2: Design modular programs using functions, arrays, and structures, and develop reusable solutions to solve real-world problems.
CO3: Demonstrate the use of pointers and dynamic memory management to analyze memory-efficient solutions for complex problems.
CO4: Construct file-based applications that enable persistent data storage and illustrate communication of results through formatted outputs.
CO5: Integrate multiple programming concepts to create a functional mini-project, demonstrating teamwork, project management skills, and adaptability to emerging challenges.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3			2						2			
CO2	2	2	3		3						2			
CO3	2	3			3						2			
CO4	2	2			3				2		2			
CO5	2	3	2	2	3			3	2	2	3			

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content:

Lab No.	Title / Experiment	Learning Focus
1	Introduction to C, Basic Input/Output, Data Types, and Operators	I/O operations, operator 1precedence, expressions
2	Problems on Conditionals: if, if-else, nested if, switch-case	Decision-making constructs
3	Looping Constructs: for, while, do-while	Iterative problem solving
4	Nested Loops: Pattern Printing, Series Problems	Logical structuring using loops
5	Functions: call by value, return types, recursion	Modular programming and recursion
6	Arrays: 1D and 2D array manipulation, search/sort problems	Data storage and iteration
7	Strings: string manipulation functions, array of strings	Character arrays and string operations
8	Pointers: pointer arithmetic, pointers with arrays and functions	Memory-level data access
9	Dynamic Memory Allocation using malloc(), calloc(), free()	Runtime memory management
10	Structures and Unions: defining, accessing, array of structures, pointer to structure.	Composite data types and access
11	File I/O: fopen(), fprintf(), fscanf(), fgetc(), fputc()	Persistent data storage
12	Mini Project: Combining structures, functions, and file I/O for a real-world scenario	Integration and application of concepts

Textbook:

1. Schaum's Outline of Programming with C by Byron S. Gottfried, McGraw-Hill Education, 1st Edition (1996)
2. Let Us C by Yashavant Kanetkar, BPB Publications, 17th Edition

Reference Books:

1. The C Programming Language by Brian W. Kernighan and Dennis M. Ritchiem, Prentice Hall, 2nd Edition
2. Mastering C by K. R. Venugopal and S. R. Prasad, Tata McGraw-Hill Education, 2nd Edition
3. Programming in ANSI C by E. Balagurusamy, McGraw Hill Education 8thEdition

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Engineering Physics Lab
Course Code: PH191
Contact: (0:0:3)
Total Contact Hours: 36
Credits: 1.5
Prerequisites: Knowledge of Physics up to 12th standard.

Course Objectives:
The aim of course is to provide adequate exposure and develop insight about the basic principles of physical sciences and its practical aspects which would help engineers to learn underlying principles of various tools and techniques they use in core engineering and related industrial applications. The course would also inculcate innovative mindsets of the students and can create awareness of the vital role played by science and engineering in the development of new technologies

Course Outcomes
After completion of this course the students will be able to

CO1: Determine mechanical properties such as Young's modulus and rigidity modulus through hands-on experiments and *analyze* material behavior under applied forces.

CO2: Perform optical experiments including Newton's Rings, laser diffraction, and optical fiber characterization, and *interpret* the results based on wave optics principles.

CO3: Investigate quantum effects such as the photoelectric effect and atomic transitions, and *relate* experimental outcomes to basic quantum principles.

CO4: Study the performance of semiconductor and electronic devices like solar cells, LEDs, and LCR circuits, and *investigate* their operational characteristics.

CO5: Conduct experiments such as Hall Effect, e/m determination, prism dispersion, or optical rotation to demonstrate the application of advanced physical principles in practical scenarios.

CO-PO Mapping

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

Course Content:

General idea about Measurements and Errors (One Mandatory):
i) Error estimation using Slide calipers/ Screw-gauge/travelling microscope for one experiment.
Experiments on Classical Physics (Any 4 to be performed from the following experiments):
1. Study of Torsional oscillation of Torsional pendulum & determination of time using various load of the oscillator.
2. Determination of Young's moduli of different materials.
3. Determination of Rigidity moduli of different materials.
4. Determination of wavelength of light by Newton's ring method.
5. Determination of wavelength of light by Laser diffraction method.
6. Optical Fibre-numerical aperture, power loss.

Experiments on Quantum Physics (Any 2 to be performed from the following experiments):
7. Determination of Planck's constant using photoelectric cell.
8. Verification of Bohr's atomic orbital theory through Frank-Hertz experiment.
9. Determination of Stefan's Constant.

10a. Study of characteristics of solar cell (illumination, areal, spectral)
10b. Study of characteristics of solar cell (I-V characteristics, Power-load characteristics, Power-wavelength characteristics)

Perform at least one of the following experiments:
11. Determination of Q factor using LCR Circuit.
12. Study of I-V characteristics of a LED/LDR.
13. Determination of band gap of a semiconductor.

**In addition, it is recommended that each student should carry out at least one experiment beyond the syllabus/one experiment as Innovative experiment.

Probable experiments beyond the syllabus:
1. Determination of the specific charge of the electron (e/m) from the path of an electron beam by Thomson method.
2. Determination of Hall co-efficient of a semiconductor and measurement of Magnetoresistance of a given semiconductor3. Study of dispersive power of material of a prism.
3. Determination of thermal conductivity of a bad/good conductor using Lees-Charlton / Searle apparatus.
4. Determination of the angle of optical rotation of a polar solution using polarimeter.
5. Any other experiment related to the theory.

Recommended Text Books for Engineering Physics Lab:

Waves & Oscillations:
1. Vibration, Waves and Acoustics- Chattopadhyay and Rakshit Classical & Modern Optics:
2. A text book of Light- K.G. Mazumder & B. Ghosh (Book & Allied Publisher)

Quantum Mechanics-I
1. Introduction to Quantum Mechanics-S. N. Ghoshal (Calcutta Book House) Solid State Physics
2. Solid State Physics and Electronics-A. B. Gupta and Nurul Islam (Book & Allied Publisher)

Text Books:
1. Practical Physics by Chatterjee & Rakshit (Book & Allied Publisher)
2. Practical Physics by K.G. Mazumder (New Central Publishing)
3. Practical Physics by R.K. Kar (Book & Allied Publisher)

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Engineering Graphics & Computer Aided Design Lab
Course Code: ME 194
Contact: 0:0:3
Credits: 1.5

Prerequisites: Basic knowledge of geometry

Course Outcomes: Upon successful completion of this course, the student will be able to:
CO1: Use common drafting tools with the knowledge of drafting standards.
CO3: Understand the concepts of engineering scales, projections, sections.
CO4: Apply computer aided drafting techniques to represent line, surface or solid models in different Engineering viewpoints.
CO5: Produce part models; carry out assembly operation and represent a design project work.

CO-PO/PSO Mapping:

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

Course Contents:
Basic Engineering Graphics:3P
Principles of Engineering Graphics; Orthographic Projection; Descriptive Geometry; Drawing Principles; Isometric Projection; Surface Development; Perspective; Reading a Drawing; Sectional Views; Dimensioning & Tolerances; True Length, Angle; intersection, Shortest Distance.

Module 1: Introduction to Engineering Drawing: 6P
Principles of Engineering Graphics and their significance, Usage of Drawing instruments, lettering, Conic sections including Rectangular Hyperbola (General method only); Cycloid, Epicycloid and Involute; Scales – Plain, Diagonal and Vernier Scales.

Module 2: Orthographic & Isometric Projections: 6P
Principles of Orthographic Projections-Conventions - Projections of Points and lines inclined to both planes; Projections of planes on inclined Planes - Auxiliary Planes; Projection of

Solids inclined to both the Planes- Auxiliary Views; Isometric Scale, Isometric Views of lines, Planes, Simple and compound Solids; Conversion of Isometric Views to Orthographic Views and Vice- versa.

Module 3: Sections and Sectional Views of Right Angular Solids: 6P
Drawing sectional views of solids for Prism, Cylinder, Pyramid, Cone and project the true shape of the sectioned surface, Auxiliary Views; Development of surfaces of Right Regular Solids - Prism, Pyramid, Cylinder and Cone; Draw sectional orthographic views of objects from industry and dwellings (foundation to slab only).

Computer Graphics:3P
Engineering Graphics Software; -Spatial Transformations; Orthographic Projections; Model Viewing; Co-ordinate Systems; Multi-view Projection; Exploded Assembly; Model Viewing; Animation; Spatial Manipulation; Surface Modeling; Solid Modeling.

Module 4: Overview of Computer Graphics: 3P
Demonstration of CAD software [The Menu System, Toolbars (Standard, Properties, Draw, Modify and Dimension), Drawing Area (Background, Crosshairs, Coordinate System), Dialog boxes and windows, Shortcut menus (Button Bars), Zooming methods, Select and erase objects].

Module 5: CAD Drawing, Customization, Annotations, layering: 6P
Set up of drawing page including scale settings, ISO and ANSI standards for dimensioning and tolerance; Using various methods to draw straight lines, circles, applying dimensions and annotations to drawings; Setting up and use of Layers, changing line lengths (extend/lengthen); Drawing sectional views of solids; Drawing annotation, CAD modeling of parts and assemblies with animation, Parametric and nonparametric solid, surface and wireframe modeling, Part editing and printing documents.

Module 6: Demonstration of a simple team design project: 3P
Illustrating Geometry and topology of engineered components: creation of engineering models and their presentation in standard 2D blueprint form and as 3D wire-frame and shaded solids; use of solid-modeling software for creating associative models at the component and assembly levels.

Text Books:
1. Bhatt N.D., Panchal V.M. & Ingle P.R, (2014), Engineering Drawing, Charotar Publishing House
2. K. Venugopal, Engineering Drawing + AutoCAD, New Age International publishers

Reference Books:
1. Pradeep Jain, Ankita Maheswari, A.P. Gautam, Engineering Graphics & Design, Khanna Publishing House

2. Agrawal B. & Agrawal C. M. (2012), Engineering Graphics, TMH Publication.
3. Shah, M.B. & Rana B.C. (2008), Engineering Drawing and Computer Graphics, Pearson Education
4. Narayana, K.L. & P Kannaiah (2008), Text book on Engineering Drawing, Scitech Publishers.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Communication and Presentation Skill
Course Code: HU191
Contact: (0:0:3)
Total Contact Hours: 36
Credit: 1.5

Pre requisites: Basic knowledge of LSRW skills.

Course Objectives: To train the students in acquiring interpersonal communication skills by focusing on language skill acquisition techniques and error feedback.

Course Outcome:
By pursuing this course, the students will be able to:

CO1: Recognize, identify and express advanced skills of Technical Communication in English and Soft Skills through Language Laboratory.
CO2: Understand, categorize, differentiate and infer listening, speaking, reading and writing skills in societal and professional life.
CO3: Analyze, compare and adapt the skills necessary to be a competent interpersonal communicator in academic and global business environments.
CO4: Deconstruct, appraise and critique professional writing documents, models and templates.
CO5: Adapt, negotiate, facilitate and collaborate with communicative competence in presentations and work-specific conclaves and interactions in the professional context.

CO-PO Mapping:

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

Course Contents:

Module 1: Introduction Theories of Communication and Soft Skills

a. Communication and the Cyclic Process of Communication (Theory, benefits and application)
b. Introduction to Workplace Communication (Principles and Practice)
c. Non-Verbal communication and its application
d. Soft Skills Introduction: Soft-Skills Introduction
e. What is Soft Skills? Significance of Soft-Skills

f. Soft-Skills Vs. Hard Skills

g. Components of Soft Skills Identifying and Exhibiting Soft-Skills (Through classroom activity)

Module 2: Active Listening

a. What is Active Listening?
b. Listening Sub-Skills—Predicting, Clarifying, Inferencing, Evaluating, Note-taking
c. Differences between Listening and Hearing, Critical Listening, Barriers to Active Listening, Improving Listening.
d. Listening in Business Telephony and Practice Practical (Role plays, case studies)

Module 3: Speaking Skills

a. Effective Public Speaking: Public Speaking, Selecting the topic for public speaking, (Understanding the audience, Organizing the main ideas, Language and Style choice in the speech, delivering the speech, Voice Clarity). Practical (Extempore)
   Self-Learning Topics: Preparation, Attire, Posture and Delivery techniques
b. Pronunciation Guide—Basics of Sound Scripting, Stress and Intonation
c. Fluency-focused activities—JAM, Conversational Role Plays, Speaking using Picture/Audio Visual inputs
d. Group Discussion: Principles, Do's and Don'ts and Practice;

Module 4: Writing and Reading Comprehension

a. Reading and Writing a Book Review (classroom activity)
b. Writing a Film Review after watching a short film (classroom activity)
c. Reading Strategies: active reading, note-taking, summarizing, and using visual aids like diagrams and graphs
d. Solving Company-Specific Verbal Aptitude papers. (Synonyms, Antonyms, Error (Correction and RC Passages)

Module 5: Presentation Skills

Kinds of Presentation. Presentation techniques, planning the presentation, Structure of presentation: Preparation, Evidence and Research, Delivering the presentation, handling questions, Time management, Visual aids.

- Self-Introduction, Creation of Video Resume`
- Need for expertise in oral presentation. Assignment on Oral presentation.
- Rules of making micro presentation (power point). Assignment on micro presentation

Text Books:

1. Pushp Lata and Sanjay Kumar. *A Handbook of Group Discussions and Job Interviews*. New Delhi: PHI, 2009.
2. Jo Billingham. *Giving Presentations*. New Delhi: Oxford University Press, 2003.
3. B. Jean Naterop and Rod Revell. *Telephoning in English*. 3rd ed. Cambridge: Cambridge University Press, 2004.
4. Jeyaraj John Sekar. *English Pronunciation Skills: Theory and Praxis*. New Delhi: Authors press, 2025.
5. Career Launcher. *IELTS Reading: A Step-by-Step Guide*. G. K. Publications. 2028

Reference Books:

1. Ann Baker. *Ship or Sheep? An Intermediate Pronunciation Course*. Cambridge: Cambridge University Press, 2006.
2. Barry Cusack and Sam McCarter. *Improve Your IELTS: Listening and Speaking Skills.* London: Macmillan, 2007.
3. Eric H. Glendinning and Beverly Holmström. *Study Reading.* Cambridge: Cambridge University Press, 2004.
4.Malcolm Goodale. *Professional Presentations*. New Delhi: Cambridge University Press, 2005.
5. Mark Hancock. *English Pronunciation in Use*. Cambridge: Cambridge University Press, 2003.
6. Tony Lynch, *Study Listening*. Cambridge: Cambridge University Press, 2004.
7. J. D. O'Connor. *Better English Pronunciation*. Cambridge: Cambridge University Press, 2005.
8. Peter Roach. *English Phonetics and Phonology: A Practical Course*. Cambridge: Cambridge University Press, 2000.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

1st Year 2nd Semester (Gr-A)									
Sl. No.	**Broad Category**	**Category**	**Paper Code**	**Subject**	**Contact Hours/Week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS201	Data Structure and Algorithms	3	0	0	3	3
2	ENGG	Minor	CS202	Introduction to Artificial Intelligence	2	0	0	2	2
3	ENGG	Major	CS203	Digital Logic and Computer Organization	3	0	0	3	3
4	SCI	Multidisciplinary	CH201	Engineering Chemistry	2	0	0	2	2
5	SCI	Multidisciplinary	M201	Engineering Mathematics–II	3	0	0	3	3
6	HUM	Value Added Course	HU202	Constitution of India & Professional Ethics	1	0	0	1	1
7	HUM	Ability Enhancement Course	HU203	Design Thinking & Innovation	1	0	0	1	1
B. PRACTICAL									
1	ENGG	Major	CS291	Data structure & Algorithms Lab	0	0	3	3	1.5
2	ENGG	Minor	CS292	Artificial Intelligence Lab	0	0	3	3	1.5
3	ENGG	Major	CS293	Digital Logic and Computer Organization Lab	0	0	3	3	1.5
4	SCI	Skill Enhancement Course	CH291	Engineering Chemistry Lab	0	0	2	2	1
5	ENGG	Skill Enhancement Course	ME293	IDEA LAB Workshop	0	0	3	3	1.5
C. MANDATORY ACTIVITIES / COURSES									
	Mandatory Course	MC281	NSS/ Physical Activities / Meditation & Yoga / Photography/		0	0	0	0	0

			Nature Club						
Total of Theory, Practical								**29**	**22**
Total First Year Credit									**40**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Data Structures and Algorithms
Course Code: CS201
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites:
1. Familiarity with the fundamentals of C or other programming language
2. A solid background in mathematics, including probability, set theory.

Course Objective(s):

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

- Gain a strong foundation in data abstraction, data types, and data structures, and understand the importance of structured data organization in solving engineering problems.
- Formulate and analyze algorithms, perform asymptotic analysis using Big O, Θ (Theta), and Ω (Omega) notations, and comprehend the trade-offs between time and space complexities.
- Design and implement linear and non-linear data structures such as arrays, linked lists, stacks, queues, trees, heaps, and graphs, and apply them effectively in computational problem-solving.
- Evaluate and compare various searching, sorting, and hashing algorithms based on their performance, and choose appropriate methods for optimized data handling.
- Appreciate the role of data structures in real-world applications, foster a mindset of lifelong learning, and develop the adaptability to utilize modern programming tools and emerging technologies.

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

CO1: Apply fundamental knowledge of data types, abstract data types, and data structures to analyze real-world computational problems and their memory/time constraints.
CO2: Design and implement linear data structures (arrays, linked lists, stacks, queues) using appropriate programming constructs to solve well-defined problems efficiently.
CO3: Develop recursive algorithms and simulate stack-based computations such as expression conversion and evaluation using appropriate engineering tools.
CO4: Construct and evaluate non-linear data structures (Binary Tree, BST, AVL Tree, heaps, graphs) and associated operations (search, insertion, deletion, traversal) to address complex engineering problems.
CO5: Compare and optimize sorting, searching, and hashing algorithms based on performance analysis and recognize their suitability in dynamic problem contexts to support life-long learning.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

CO-PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO 3
CO1	3	3												
CO2	3	2	3		3									
CO3	2	2	3		3									
CO4	3	3	2	3	3									
CO5	3	3			2						3			

Course Content:

Module 1: Introduction 4L

Concepts of data and information; Concept of Abstract Data Type, Data Structure and Data Type. Classification of Data Structures- Primitive and Non-Primitive Data Structure, Linear and Non-Linear Data Structure. Need of Data Structures.

Concept of algorithms and programs, Different methods of representing algorithm; Algorithm analysis, time and space analysis of algorithms – Asymptotic notations like Big Oh (O), Small Oh(o), Big Omega(Ω), Small Omega(ω) and Theta(Θ) notation (definition and significance).

Module 2: Non-Restricted Linear Data Structure 9L

List or Linear List: Definition and Example, List as ADT. Representation of Linear List- Sequential Representation and Linked Representation.

Array: Introduction to sequential representation, Linearization of multidimensional array. Application of array- representation of polynomial using array, Representation of Sparse matrix using array.

Linked List: Introduction to linked representation, Implementation of different types of linked list- Singly linked list, Doubly linked list, Circular linked list, Circular Doubly Linked List. Application of Linked list- Representation of polynomial.

Module 3: Restricted Linear Data Structure 6L

Stack: Definition of Stack, implementations of stack using array and linked list

Applications of stack- infix to postfix conversion, Postfix Evaluation

Recursion: Principles of recursion - use of stack, tail recursion. Tower of Hanoi using recursion.

Queue: Definition of Queue; Implementation of queue using array-physical, linear and circular model; Implementation of queue using linked list.

Dequeue - Definition and different types of dequeue.

Module 4: Nonlinear Data structures 9L

Trees and Binary Tree:

Basic terminologies; Definition of tree and binary tree. Difference between tree and binary tree, Representation of binary tree (using array and linked list)

Binary tree traversal (pre-, in-, post- order); Threaded binary tree- definition, insertion and deletion algorithm; Binary search tree- Definition, insertion, deletion, searching algorithm;

Height balanced binary tree: AVL tree- definition, insertion and deletion with examples only.

m –Way Search Tree: B Tree – Definition, insertion and deletion with examples only; B+ Tree – Definition, insertion and deletion with examples only.

Heap: Definition (min heap and max heap), creation, insertion and deletion algorithm. Application of heap (priority queue and sorting).

Graphs: Definition and representation (adjacency matrix, incidence matrix and adjacency list).

Graph traversal– Depth-first search (DFS), Breadth-first search (BFS) - concepts of edges used in DFS and BFS (tree-edge, back-edge, cross-edge, and forward-edge).

Module 5: Sorting and Searching 8L

Sorting Algorithms: Definition and need of sorting, different types of sorting algorithm (internal, external, stable, in-place, comparison based); Factors affecting sorting Methods, Bubble sort, Insertion sort, Selection sort, Quick sort, Merge sort, Radix sort – algorithm with analysis (time complexity)

Searching: Factors affecting searching Methods; Sequential search –algorithm with analysis (time complexity); improvement using sentinel.

Binary search and Interpolation Search algorithm with analysis (time complexity)

Hashing: Introduction and purpose of Hashing and Hash functions (division, folding and mid-square), Collision resolution techniques.

Text book:

1. Data Structures Through 'C' Language by Samiran Chattopadhyay, Debabrata Ghosh Dastidar, Matangini Chattopadhyay, Edition: 2001, BPB Publications
2. Fundamentals of Data Structures of C by Ellis Horowitz, Sartaj Sahni, Susan Anderson-freed 2nd Edition, Universities Press

Reference Books:

1. Data Structures, Algorithms, and Software Principles in C by Thomas A. Standish, 1 Edition, Pearson.
2. Data Structures by S. Lipschutz, Special Indian Edition, Tata McGraw Hill Education (India) Private Limited
3. Data Structures and Program Design in C by Robert L. Kruse, Bruce P. Leung 2nd Edition, Pearson
    1. Data Structures in C by Aaron M. Tenenbaum, 1St Edition, Pearson

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name:	**Introduction to Artificial Intelligence**
Course Code:	**CS202**
Contact Hours (Period/week):	**2**
Total Contact Hours:	**30**
Credit:	**2**

Course Objectives:

The objectives of this course are to enable students to

1. Comprehend the fundamental concepts of Knowledge Representation and Inferencing in Artificial Intelligence and its utilitarian importance in current technological context.
2. Formulate a problem as State-Space Exploration Framework or an Inferencing Framework of Artificial Intelligence.
3. Use the strategies of AI-Heuristics to find acceptable solutions avoiding brute- force techniques.
4. Design AI-Frameworks for Inferencing based on knowledge base.
5. Analyze the effectiveness of AI-Inferencing Model in offering solutions to the respective problem.

Course Outcomes (COs):

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

CO1	Understand and explain the fundamental concepts of Knowledge Representation and Inferencing in Artificial Intelligence and its utilitarian importance in current technological context for further exploration leading towards lifelong learning.
CO2	Identify and formulate an engineering problem primarily to fit a State-Space Exploration Framework or an Inferencing Model/Agent Design Framework within the scope of Artificial Intelligence paradigm.
CO3	Explore relevant literature and apply the concept of Heuristic Techniques of Artificial Intelligence to solve problems.
CO4	Develop Inferencing Models for proposing solutions to the problems of Artificial Intelligence.
CO5	Implement Inferencing Models of Artificial Intelligence through developing feasible algorithms and investigate their effectiveness by analyzing their performances in solving the relevant problems.

CO–PO Mapping:

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CO1	3	2	-	-	-	-	-	-	-	-	3	3	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-	-	3	-
CO3	2	2	3	2	-	-	-	-	-	-	-	2	-	2
CO4	2	2	2	3	-	-	-	-	-	-	2	2	-	2
CO5	2	2	3	3	2	-	-	-	-	-	2	2	2	3

Course Contents:

Module 1: Introduction to Artificial Intelligence (3L)
Why AI • Definition of AI • Goals of AI • History and evolution of AI • Types of AI: Narrow, General, Super • Human vs Artificial Intelligence • Applications of AI in various domains • AI for social good

Module 2: Intelligent Agents and Logic-Based Thinking (8L)
Intelligent systems • Agents and environments • Decision making using rules and logic • Symbolic AI concepts • Propositional Logic: Knowledge Representation and Inference using Propositional Logic • Predicate Logic: Knowledge Representation, Inference and Answer Extraction using First Order Predicate Logic

Module 3: Overview of AI Branches and Perception (8L)
Machine learning • Deep learning • Natural language processing • Computer vision • Expert systems • Fuzzy logic • Evolutionary algorithms • Reinforcement learning • Planning and scheduling • Human-AI collaboration

Module 4: Basics of Machine Learning (6L)
What is machine learning • AI vs ML • Types of learning: supervised, unsupervised • Concept of dataset, features, and labels • ML model and prediction flow • Common ML applications Introduction to decision trees (concept only) • ML pipeline overview.

Module 5: Applications and Ethics of AI (5L)
AI in robotics and automation • AI-enabled smart applications • Industry 4.0 and intelligent systems • AI in different sectors: healthcare, agriculture, transport, education, etc. • Human-AI teamwork • Basics of AI ethics: bias, fairness, privacy • Career opportunities and future scopes in AI.

Textbook:

1. Saptarsi Goswami, Amit Kumar Das, Amlan Chakrabarti – AI for Everyone: A Beginner's Handbook for Artificial Intelligence (AI), Pearson.
2. Rich, E., Knight, K and Shankar, B. 2009. Artificial Intelligence, 3rd edition, Tata McGraw Hill.
3. Russell, S. and Norvig, P. 2015. Artificial Intelligence - A Modern Approach, 3rd edition, Prentice Hall.

Reference Books:

1. Reema Thareja, Artificial Intelligence: Beyond Classical AI, Pearson.
2. Patterson, Introduction to Artificial Intelligence and Expert Systems, Pearson.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Digital Logic and Computer Organization
Course Code: CS203
Contact Hours: 3:0:0
Total Contact Hours: 36
Credits: 3

Course Objectives:
By the end of this course, students will be able to:

- To introduce number systems, logic gates, and design of combinational and sequential circuits.
- To develop an understanding of data processing using micro-operations and instruction formats.
- To explain how CPU, memory, and I/O units are organized and interact during instruction execution.
- To describe arithmetic algorithms and control unit designs in processor architecture.
- To build a foundation for advanced topics like microprocessors, computer architecture, and embedded systems.

Course Outcomes (COs):
After successful completion of the course the students will be able to

CO1	Explain various number systems and coding schemes, and apply Boolean algebra laws and Karnaugh Maps to simplify logical expressions.
CO2	Design and construct combinational and sequential logic circuits including adders, multiplexers, flip-flops, and counters for implementing digital functions.
CO3	Develop and analyze data path units such as ALU, control units, and register organizations to support instruction execution in CPU architectures.
CO4	Demonstrate arithmetic operations like Booth's multiplication and division, and illustrate various addressing modes and instruction formats used in CPUs.
CO5	Compare memory hierarchy systems and I/O techniques, and evaluate their role in enhancing overall processor performance.

CO-PO Mapping:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3			2									
CO2	3	3	3		3									
CO3	3	2			3									
CO4	3	2			2									
CO5	3	2			2					2	3			

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content

Module 1: Number Systems, Boolean Algebra, and Logic Simplification (6 L)

- Binary, BCD, ASCII, EBCDIC, Gray Code & conversions [1L]
- Boolean Algebra – Laws, Theorems [1L]
- Boolean Functions, Minterm & Maxterm, SOP & POS Forms [2L]
- Karnaugh Map (up to 4-variable), Algebraic Simplification [2L]

Module 2: Combinational Circuits (6 L)

- Half & Full Adder/Subtractor, Serial & Parallel Adders, CLA Adder [2L]
- Parity Generator, Encoder, Decoder, Multiplexer, Demultiplexer [2L]
- Comparator, Code Converters [2L]

Module 3: Sequential Circuits & Registers (6 L)

- Flip-Flops: SR, JK, Master-Slave JK, D, T; Characteristic & Excitation Tables [2L]
- Counters: Synchronous/Asynchronous, Ring & Johnson, Mod-N Counters [2L]
- Registers: SISO, SIPO, PIPO, PISO [1L]
- Applications of Counters and Registers [1L]

Module 4: Data Representation & Arithmetic Operations (5 L)

- Integer Arithmetic (Add, Subtract), Booth's Multiplication Algorithm [2L]
- Restoring & Non-Restoring Division [1L]
- Instruction Formats and Addressing Modes [2L]

Module 5: CPU and Control Unit Organization (6 L)

- Register Transfer Language (RTL), Bus Architecture, Micro-operations [1L]
- ALU Design, Status Flags, General Register & Stack Organization [2L]
- Control Unit: Hardwired vs. Microprogrammed Control, Sequencing [2L]
- Basic Instruction Cycle and Execution Pipeline [1L]

Module 6: Memory & I/O Organization (7 L)

- RAM, ROM Types, Memory Hierarchy: Cache, Main, Secondary [1L]
- Cache Mapping: Direct, Associative, Set-Associative; Write Policies [3L]
- Virtual Memory: Paging, Segmentation, FIFO & LRU [1L]
- I/O Transfer Modes: Programmed I/O, Interrupt-Driven I/O, DMA [1L]
- Interrupts: Maskable/Non-Maskable, Daisy Chaining; I/O Processor [1L]

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Textbooks:

1. Digital Logic and Computer Design by M. Morris Mano, Pearson Education, 1st Edition
2. Computer Organization and Architecture: Designing for Performance by William Stallings, Pearson Education, 10th Edition

Reference Books:

1. Digital Design by M. Morris Mano, Michael D. Ciletti, Pearson Education, 5th Edition
2. Computer Organization and Embedded Systems by Carl Hamacher, Zvonko Vranesic, Safwat Zaky, McGraw-Hill Education, 6th Edition
3. Computer Organization and Design: The Hardware/Software Interface by David A. Patterson, John L. Hennessy, Morgan Kaufmann Publishers, RISC-V Edition
4. Fundamentals of Logic Design by Charles H. Roth Jr., Larry L. Kinney, Cengage Learning, 7th Edition
5. Digital Fundamentals by *Thomas L. Floyd,* Pearson Education, 11th Edition

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: ENGINEERING CHEMISTRY
Paper Code: CH 201
Total Contact Hours: 24
Credit: 2
Prerequisites: 10+2

COURSE OBJECTIVE

- Understand the basic principles of atomic structures and periodic properties of elements, different engineering materials, advanced polymers.
- Apply the knowledge of free energy, energy storage device and semiconductors to design environment friendly and sustainable devices.
- Apply the concept of corrosion and fuel to improve its efficacy and application for industrial purpose.
- Analyze the organic reaction with the structure of organic molecules by applying the knowledge of different spectroscopic techniques.
- Evaluate the electrical, optical, and structural properties of semiconductors to analyze their potential applications in modern electronic and energy devices

COURSE OUTCOME

After completion of this course the students will be able to

CO1. Understand the basic principles of atomic structures and periodic properties of elements, different engineering materials, advanced polymers.
CO2. Apply the knowledge of free energy, energy storage device and semiconductors to design environment friendly and sustainable devices.
CO3. Utilize the concept of corrosion and fuel to improve its efficacy and application for industrial purpose.
CO4. Analyze the organic reaction with the structure of organic molecules by applying the knowledge of different spectroscopic techniques.
CO5. Evaluate the electrical, optical, and structural properties of semiconductors to analyze their potential applications in modern electronic and energy devices

CO-PO mapping:

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

COURSE CONTENT

Module 1:

Quantum Properties of Atoms (4L)
Schrodinger Wave Equation (time independent – basic principles only), de Broglie Equation, Heisenberg Uncertainty Principle, Quantum Numbers, Effective nuclear charge, Slater's rule, penetration of orbitals, variations of orbital energies in the periodic table, atomic and ionic sizes, ionization energies, electron affinity and electronegativity, oxidation properties.

Chemistry of materials (2L)
Semiconductor-Based Memory Materials (Si & Ge) [Introduction, Properties and role of Si & Ge), Intensive & Extensive semiconductor

Module II:

Chemical Thermodynamics (5L)

1^{st} & 2^{nd} Law of Thermodynamics, Tendency for maximum randomness, Carnot Heat Engine [Derivation], Entropy characteristics, Mathematical explanation & physical significance of Entropy, Entropy change of ideal gas for isothermal reversible process, Gibbs free Energy Function, Standard free Energy, Criterion of spontaneity.

Electricity production through chemical reactions (2L)

Electrochemical Cell, writing of cell notation, free energy and EMF, Criterion of spontaneity in terms of Cell,

Nernst equation (only expression, no derivation) and applications, calculation of EMF of a cell, calculation of single electrode potential, calculation of Kc, calculation of Kc from G^0.

Working principle and applications of Lithium-ion batteries

Module III:

Polymers for Engineering Applications (3L)

Polymers and their classifications (based on origin, chemical structure, polymeric structure, tacticity and molecular forces)

Commercially important polymers: Synthesis and applications of Bakelite, nylon 6,6, HDPE & LDPE

Conducting polymers –Types examples and applications.

Biodegradable polymers –definition, example and uses

Industrial Chemistry (3L)

Types of corrosion, Electrochemical theory of corrosion, rusting of iron, comparison of chemical & electrochemical corrosion. [Mechanism excluded]

Factors affecting the rate of corrosion; nature of metal (physical state, purity, position in Galvanic series) & environment.

Corrosion control: Cathodic protection, anodic protection, Inorganic coatings.

Classification of Fuel (LPG, CNG, BIOGAS), Calorific value, Octane number, Cetane number, HCV, LCV. [Definition only]

Module IV:

Organic Reactions & synthesis of drugs (3L)

Acidity and basicity comparison of organic compounds (acids, alcohols & amines), Nucleophilic Substitution reaction and Electrophilic Addition reactions, Markonikov's rule, peroxide effect, Synthesis of Paracetamol& Aspirin and uses. (Name reactions are not in syllabus)

Spectroscopy (2L)

Electromagnetic spectrum, Lambert-Beer Law, Finding of λ max value & concentration of the unknown solution, Applications of UV-VIS spectroscopy, Chromophores & Auxochromes.
Applications of IR spectroscopy, Fingerprint region

Text Books:

1. Chemistry –I, Gourkrishna Das Mohapatro
2. A text book of Engineering Chemistry, Dr. Rajshree Khare
3. Engineering Chemistry, U. N. Dhar
4. Physical Chemistry, P.C. Rakshit

Reference Books:

1. Engineering Chemistry, Jain & Jain
2. Engineering Chemistry (NPTEL Web-book), by B. L. Tembe, Kamaluddin and M. S.Krishna
3. Text book of Engineering Chemistry, Jaya Shree Anireddy

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Engineering Mathematics - II
Paper Code: M 201
Contact (L: T: P): 3 : 0 : 0
Total Contact Hours: 36
Credit: 3

Prerequisites:

The students to whom this course will be offered must have the understanding of (10+2) standard algebraic operations, and elementary calculus concepts including limits, continuity, differentiation, and integration.

Course Objectives:

The objective of this course is to familiarize the prospective engineers with techniques in ordinary differential equations, Laplace transform and numerical methods. It aims to equip the students with standard concepts and tools at an intermediate to advanced level that will serve them well towards tackling more advanced level of mathematics and applications that they would find useful in their disciplines.

Course Outcomes (COs):

On successful completion of the learning sessions of the course, the learner will be able to:

CO1. Apply analytical methods to solve ordinary differential equations in engineering contexts.
CO2. Apply the properties and inverse of Laplace Transforms to compute improper integrals and determine solutions of linear ordinary differential equations with constant coefficients in engineering scenarios.
CO3. Apply numerical methods to interpolate data, perform numerical integration, and solve ordinary differential equations in engineering applications.
CO4. Analyze the behavior of solutions using analytical and numerical approaches, including Laplace transforms, to assess stability, convergence, and accuracyin engineering contexts.

CO-PO/PSO Mapping:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	-	-	-	-	-	-	-	-	1
CO2	3	2	-	-	-	-	-	-	-	-	1
CO3	3	2	-	-	-	-	-	-	-	-	1

CO4	3	3	1	1	-	-	-	-	-	-	2
M 201	**3**	**2.25**	**1**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**1.25**

Weightage Values: Strongly mapped: '3', Moderately mapped: '2', Weakly mapped: '1', Not mapped: '-'.

Course Content:

Module I: First Order Ordinary Differential Equations (ODE) (9L)
Solution of first order and first-degree ODE: Exact ODE, Rules for finding Integrating factors, Linear ODE, Bernoulli's equation.
Solution of first order and higher degree ODE: solvable for$p$,solvable for y and solvable for $x$and Clairaut's equation.

Module II: Second Order Ordinary Differential Equations (ODE) (8L)
Solution of second order ODE with constant coefficients: Complementary Function and Particular Integral, Method of variation of parameters, Cauchy-Euler equations.

Module III: Laplace Transform (LT) (12L)
Concept of improper integrals; Definition and existence of LT, LT of elementary functions, First and second shifting properties, Change of scale property, LT of $tf(t)$, LT of $\frac{f(t)}{t}$, LT of derivatives of $f(t)$, LT of integral of $f(t)$, Evaluation of improperintegrals using LT, LT of periodic and step functions, Inverse LT: Definition and its properties,Convolution theorem (statement only) and its application to the evaluation of inverse LT, Solution oflinear ODE with constant coefficients (initial value problem) using LT.

Module IV: Numerical Methods (7L)
Introduction to error analysis, Calculus of finite difference. **Interpolation:** Newton forward and backward interpolation, Lagrange's interpolation. **Numerical integration:** Trapezoidal rule, Simpson's $1/3$ Rule. **Numerical solution of ordinary differential equation:** Euler method, Fourth order Runge-Kutta method.

Text Books:

1. Grewal, B.S., Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.
2. Kreyszig, E., Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.

Reference Books:

1. Guruprasad, S. A text book of Engineering Mathematics-I, New age International Publishers.

2. Ramana, B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
3. Veerarajan, T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
4. Bali, N.P. and Goyal, M., A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
5. Thomas, G.B. and Finney, R.L., Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
6. Apostol, M., Calculus, Volumes 1 and 2 (2nd Edition), Wiley Eastern, 1980.
7. Kumaresan, S., Linear Algebra - A Geometric approach, Prentice Hall of India, 2000.
8. Poole, D., Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005.
9. Bronson, R., Schaum's Outline of Matrix Operations. 1988.
10. Piskunov, N., Differential and Integral Calculus, Vol. I & Vol. II, Mir Publishers, 1969.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Paper Name: **Constitution of India and Professional Ethics**
Paper Code: HU202
Contact: 1:0:0
Credit: 1
Total Lectures: 12
Prerequisites:
Basic knowledge (10+2 level) of the Indian Constitution and moral science.

Course outcome: On completing this course the student will be able to

CO1: Identify, define and understand the significance of the Constitution of India, its spirit and values and the fundamental rights and duties as a responsible citizen.
CO2: define and discover core ethical concepts, the basic perception of profession, and professional ethics that shape the ethical behavior of an engineer.
CO3: identify, examine and apply codes of engineering ethics, engineers' social responsibilities and industrial standards and ethical dilemmas.
CO4: consider, correlate and appraise ethical leadership and principles in addressing gender issues, concerns of IPR and industrial responsibilities.

Module 1: [2L]
Preamble: Salient Features, Fundamental Rights, Fundamental Duties, Directive Principles of State Policy, Parliament -Powers and Functions –Executive- President -Governor - Council of Ministers.

Module 2: [**3L]**
Introduction to Ethical Thinking; what is Ethics, Work ethics; Scope of Professional Ethics, Values and Characteristics, Types of values: Negative and positive values, Ethical values for Professional success.

Module 3: [4L]

Engineering Ethics, Ethical theories: a brief overview; utilitarianism, deontology, virtue ethics.
Professional Codes, Codes of professional ethics-Moral dilemmas, and moral autonomy-Internal ethics of business: whistle blowing, conflicts of interest, Job discrimination, and Exploitation of Employees; Social and ethical responsibilities of technologists: Responsibilities towards Customers, shareholders, employees – Social Audit.

Case Studies: Bhopal Gas Tragedy, Chernobyl (linking ethics to real-world failures).

Module 4: [**3L]**
Business ethics, ethical decision-making frameworks - Impact of ethics on business policies and strategies- Characteristics of ethical leaders; fostering integrity in teams; Addressing occupational crime, discrimination, and gender-based issues in workplaces-Intellectual

property rights (IPR), Plagiarism and Academic Misconduct.

Text Books:

1. Durga Das Basu. *Introduction to the Constitution of India*. 27th ed. New Delhi: Lexis Nexis, 2024.
2. R.S Nagarajan. *A Textbook on Professional Ethics and Human Values*. New Age International (P) Limited, 2022.
3. N. Subramanian. *Professional Ethics.* New Delhi: Oxford University Press, 2017.
4. A N Tripathi, *Human Values*. New Delhi: New Age Publishers, 2019.
5. S. K. Chakraborty. *Values and Ethics for Organizations: Theory and Practices*. New Delhi: Oxford University Press, 1997.

Reference Books:

1. O. C. Ferrell, John Friaedrich and Linda Ferrell. *Business Ethics: Ethical Decision Making and Cases.* New Delhi: Cengage India, 2024.
2. Charles Fledderman. *Engineering Ethics*. 3rd ed. New Delhi: Pearson Education, 2007.
3. Dinesh G. Harkut and Gajendra R. Bamnote. *Professional Ethics for Engineers*. Chennai: Notion Press, 2023.
4. U.C. Mathur, *Corporate Governance and Business Ethics*: *Text and Cases*. Chennai: Macmillan, 2012.
5. Fernando. A. C., K. P. Muralidheeran and E. K. Satheesh. *Business Ethics – An Indian Perspective*. New Delhi: Pearson Education, 2019.

CO-PO Mapping:

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**
CO 1	-	-	-	-	-	-	**3**	**2**	-	-	**2**
CO 2	-	-	-	-	-	-	**3**	**2**	-	-	**2**
CO 3	-	-	-	-	-	**2**	**3**	**2**	-	-	**2**
CO 4	-	-	-	-	-	**2**	**3**	**3**	-	-	**2**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Title	**Design Thinking and Innovation**		
Course Code	**HU203**		
(L-T-P)	**(2-0-0)**		
Class Hours/Week	**02**		
Total class hours	**30**		

Course Objective:

The objective of this Course is to provide new ways of creative thinking and learn the innovation cycle of Design Thinking process for developing innovative products and services which are useful for a student in preparing for an engineering career.

Course Outcomes (COs): Upon completion of the course, students shall be able to

Sl. No.	Course outcomes	Mapping to POs
1.	Analyze emotional experience and expressions to better understand stakeholders while designing innovative products through group brainstorming sessions.	PO1, PO2, PO4, PO5, PO7, PO8 & PO9
2.	Generate and develop design ideas through different technique	PO1, PO2, PO3, PO4, PO5, PO7, PO8, PO10 & PO11
3.	Develop new ways of creative thinking and learn the innovation cycle of Design Thinking process for developing any innovative products using facility in AICTE IDEALAB	PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO10 & PO11

CO-PO MAPPING:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	2	-	2	2	-	2	3	1	-	-
CO2	1	2	3	3	3	-	2	3	-	3	2
CO3	1	3	3	3	3	2	2	3	-	2	2

Prerequisites:

For a course on the Basics of Design Thinking, students should ideally possess basic computer skills, communication abilities, problem-solving aptitude, critical thinking, introductory knowledge of Sustainable Development Goals, curiosity, and openness to new ideas, as well as basic understanding of mathematics, technology, and manufacturing processes.

However, even if these prerequisites are not satisfied, the faculty will cover them in the first few classes.

An awareness of 21st-century skills, including creativity and collaboration, is also beneficial.

These prerequisites aim to provide a foundation, and any gaps in knowledge will be addressed by the instructor early in the course.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

SYLLABUS:

Module	Content	Hour
Module 1:	**Basics of Design Thinking:** Definition of Design Thinking, Need for Design Thinking, history of Design Thinking, Concepts & Brainstorming, 2X2 matrix, 6-3-5 method, NABC method;	3
Module 2:	**PROCESSOFDESIGN: Understanding Design thinking** Sharedmodelinteam-baseddesign–TheoryandpracticeinDesignthinking–Explorepresentationsigners across globe– MVPor Prototyping. Stages of Design Thinking Process (explain with examples) – Empathize (Methods of Empathize Phase: Ask 5 Why / 5W+H questions, Stakeholder map, Empathy Map, Peer observation, Trend analysis). Define (Methods of Define Phase: Storytelling, Critical items diagram, Define success). Ideate (Brainstorming, 2X2 matrix, 6-3-5 method, NABC method). Prototype (Types of prototypes - Methods of prototyping - Focused experiments, Exploration map, Minimum Viable Product). Test (Methods of Testing: Feedback capture grid, A/B testing).	6
Module 3:	**Tools for Design Thinking** Real-Timedesigninteractioncapturesandanalysis–Enablingefficientcollaborationindigitalspace–Empathyfordesign–CollaborationindistributedDesign	3
Module 4:	**Design Thinking in IT** Design Thinking to Business Process modelling – Agile in Virtual collaboration environment – Scenario based Prototyping	2
Module 5:	**Design Thinking for strategic innovations** Growth – Story telling representation – Strategic Foresight - Change – Sense Making - MaintenanceRelevance–Valueredefinition-ExtremeCompetition–experiencedesign-Standardization–Humanization-CreativeCulture–Rapid prototyping, Strategy and Organization–Business Model	3
Module 6:	**Problem Solving & Critical thinking** Introduction to TRIZ, SCAMPER, UI and UX.	2
Module 7:	**Sustainable development goals (SDG)** Integrating and mapping 17 Sustainable development goals (SDG) during designing a product; goods or service. Introduction to 21st Century Skill Set	1
Module 8:	**Case Study & Project Report Submission**	10

Text Books:

1. Karmin Design Thinking by Dr. Bala Ramadurai, Mudranik Technology Private Ltd. ISBN 978-93-5419-0I0-0.
2. John.R.Karsnitz, Stephen O'Brien and John P. Hutchinson, "Engineering Design", Cengage learning (International edition) Second Edition, 2013.
3. Roger Martin, "The Design of Business: Why Design Thinking is the Next Competitive Advantage", Harvard Business Press, 2009.
4. Hasso Plattner, Christoph Meinel and Larry Leifer (eds), "Design Thinking: Understand – Improve – Apply", Springer, 2011
5. Idris Mootee, "Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School", John Wiley & Sons 2013.

Reference Books:

1. Yousef Haik and Tamer M.Shahin, "Engineering Design Process", Cengage Learning, Second Edition, 2011.
2. Solving Problems with Design Thinking - Ten Stories of What Works (Columbia Business School Publishing) Hardcover – 20 Sep 2013 by Jeanne Liedtka (Author), Andrew King (Author), Kevin Bennett (Author).
3. Tim Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, HarperCollins e-books, 2009.
4. Michael Lewrick, Patrick Link, Larry Leifer, The Design Thinking Toolbox, John Wiley & Sons, 2020.
5. Michael Lewrick, Patrick Link, Larry Leifer, The Design Thinking Playbook, John Wiley & Sons, 2018.
6. Kristin Fontichiaro, Design Thinking, Cherry Lake Publishing, USA, 2015.
7. Walter Brenner, Falk Uebernickel, Design Thinking for Innovation - Research and Practice, Springer Series, 2016.
8. Gavin Ambrose, Paul Harris, Design Thinking, AVA Publishing, 2010.
9. Muhammad Mashhood Alam, Transforming an Idea into Business with Design Thinking, First Edition, Taylor and Francis Group, 2019.
10. S. Balaram, Thinking Design, Sage Publications, 2011.

WEB REFERENCES:

1. https://designthinking.ideo.com/
2. https://thinkibility.com/2018/12/01/engineering-vs-design-thinking/
3. https://www.coursera.org/learn/design-thinking-innovation
4. https://swayam.gov.in/nd1_noc20_mg38/preview
5. www.tutor2u.net/business/presentations/. /productlifecycle/default.html
6. https://docs.oracle.com/cd/E11108_02/otn/pdf/. /E11087_01.pdf
7. www.bizfilings.com › Home › Marketing › Product Developmen
8. https://www.mindtools.com/brainstm.html

9. https://www.quicksprout.com/. /how-to-reverse-engineer-your-competit
10. www.vertabelo.com/blog/documentation/reverse-engineering
https://support.microsoft.com/en-us/kb/273814
11. https://support.google.com/docs/answer/179740?hl=en

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Data structure &Algorithms Lab
Course Code: CS291
Contact (Periods/Week):3P/Week
Total Contact Hours: 36
Credits:1.5

Course Objectives
By the end of this course, students will be able to:

- To develop skills in implementing and analyzing data structures using C.
- To gain hands-on experience in solving problems using arrays, linked lists, stacks, queues, trees, graphs, and hashing.
- To apply algorithmic concepts like recursion, sorting, and searching in solving real-world problems.

Course Outcomes (COs):
After successful completion of the course the students will be able to

CO1	Apply fundamental programming concepts to construct and manipulate linear data structures like arrays, linked list, stacks, and queues for solving structured problems.
CO2	Develop and analyze non-linear data structures such as binary search trees to address hierarchical and dynamic memory-based problems.
CO3	Implement recursive algorithms to solve classical problems like Tower of Hanoi and Fibonacci series, demonstrating critical thinking and abstraction.
CO4	Compare and evaluate various sorting and searching algorithms based on time and space complexity for performance-critical applications.
CO5	Design and integrate suitable data structures to build efficient software modules, demonstrating teamwork, project planning, and communication of technical results.

CO-PO-PSO Mapping:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3			3									
CO2	3	2			3									
CO3		2		3	2									
CO4		3	3		3									
CO5		2	3		3			2	2	2	3			

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content:

Lab No.	Title	Topics / Experiments
1	Introduction to C Revisions	Basic C programming constructs, functions, pointer concepts.
2	Arrays and Polynomial Representation	Create, access and manipulate 1D, 2D arrays; polynomial representation using arrays.
3	Linked Lists	Singly Linked List: creation, insertion, deletion, search.
4	Doubly & Circular Linked Lists	Implement doubly linked and circular linked list with insertion/deletion.
5	Stacks (Array & Linked List)	Implement stack using array and linked list.
6	Application of Stack	infix to postfix conversion, postfix evaluation.
7	Queues (Array & Linked List)	Physical, Linear and circular model of queues using array, Queue Using linked list.
8	Recursion Applications	Factorial, Fibonacci, Tower of Hanoi.
9	Binary Search Tree (BST)	Insertion, deletion, searching; height of tree.
10	Sorting Algorithms	Implement of bubble sort, insertion sort, and selection sort.
11	Sorting Algorithms	Implement of quick, merge sort, and radix sort.
12	Searching and Hashing	Linear search, binary search, interpolation search;

Textbooks:

1. Data Structures Through C Language by Samiran Chattopadhyay, Debabrata Ghosh Dastidar, Matangini Chattopadhyay, Edition: 2001, BPB Publications.
2. Fundamentals of Data Structures of C by Ellis Horowitz, Sartaj Sahni, Susan Anderson-freed 2nd Edition, Universities Press.

Reference books:

1. Data Structures, Algorithms, and Software Principles in C by Thomas A. Standish, 1 Edition, Pearson.
2. Data Structures by S. Lipschutz, Special Indian Edition, Tata McGraw Hill Education (India) Private Limited
3. Data Structures and Program Design in C by Robert L. Kruse, Bruce P. Leung 2nd Edition, Pearson.
4. Data Structures in C by Aaron M. Tenenbaum, 1st Edition, Pearson

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name:	**Artificial Intelligence Lab**
Course Code:	**CS292**
Contact Hours (Period/week):	**3P/Week**
Total Contact Hours:	**30**
Credit:	**1.5**

Course Objectives:

The objectives of this course are to enable students to

1. Gain foundational knowledge of PROLOG to implement an Artificial Intelligent Agent as an executable computer program for Knowledge Representation and Inferencing
2. Formulate a problem by analyzing its characteristics to fit a State-Space Exploration Framework or an Inferencing Framework of Artificial Intelligence.
3. ApplytheconceptsofArtificialIntelligencetosolveaproblembyimplementing well-known Artificial Intelligence strategies using proper techniques and tools of PROLOG.
4. BuildexpertsystemsofferingsolutionstothechallengingproblemsofArtificial Intelligence.
5. ImplementArtificialIntelligencebasedideasasexecutablePROLOGprograms through developing intelligent heuristic strategies

Course Outcomes (COs):

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

CO1	Acquire foundational knowledge of PROLOG to implement an Artificial Intelligent Agent as an executable computer program for Knowledge Representation and Inferencing and understand the working principle of the agent and assess its utilitarian importance in current technological context leading towards lifelong learning.
CO2	Identify and formulate an engineering problem by analyzing its characteristics to fit a State-Space Exploration Framework or an Inferencing Agent Formulation Framework of Artificial Intelligence.
CO3	Explore relevant literature and apply the concepts of Artificial Intelligence to solve a problem by implementing well- known Artificial Intelligence strategies using proper techniques and tools of PROLOG.
CO4	Develop ideas and propose an expert system offering solutions to the challenging problems of Artificial Intelligence.

CO5	Plan and Implement Artificial Intelligence based ideas as executable PROLOG programs through developing intelligent heuristic strategies or expert systems with adequate documentation in a collaborative environment for successfully carrying out projects on Artificial Intelligence Problems and investigate their effectiveness by analyzing the performances using proper techniques and tools.

Course Contents:

Module 1: Introduction to PROLOG Programming along with the IDE and its Basic Components
Assignments for understanding the Basic Components of Knowledge Representation and Inferencing in Artificial Intelligence using PROLOG Programming and its working strategy. Understanding facts, rules, queries, and syntax.

Module2: Recursive definitions in Prolog
Fibonacci Series, Calculator, Factorial, summation, list length, etc. Using recursive rules.

Module3: Defining facts and simple queries
Writing a knowledge base for family relationships, basic objects.

Module4: **Rules and inference in Prolog**
Creating logical rules and testing inferences.

Module5: List operations in Prolog
Checking membership, concatenation, reverse, max/minoflist.

Module6: Pattern matching and symbolic reasoning
Simple examples involving pattern recognition (e.g., shape or name matching, Family Tree design)

Module7: Expert system simulation (Mini project)
Building a mini knowledge-based system (e.g., Animal Classification, Medical diagnosis, etc.).

Textbook:

1. Ivan Bratko, Prolog Programming for Artificial Intelligence,4thEdition, Addison-Wesley.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	3	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-	-	3	-
CO3	2	2	3	2	-	-	-	-	-	-	-	2	-	2
CO4	2	2	2	3	-	-	-	-	-	-	-	2	-	2
CO5	2	2	3	3	2	2	2	2	2	2	2	2	2	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Digital Logic and Computer Organization Lab
Course Code: CS293
Contact Hours: 0:0:3
Total Contact Hours: 36
Credits:1.5

Course Objectives
By the end of this course, students will be able to:

- To provide hands-on experience in designing and analyzing combinational and sequential logic circuits.
- To enhance understanding of digital systems using simulation and HDL tools.
- To familiarize students with arithmetic circuits, memory design, and basic CPU control logic through practical implementation.

Course Outcomes (COs):
After successful completion of the course the students will be able to

CO1	Implement and verify the functionality of basic and derived logic gates, using ICs and simulation tools to demonstrate fundamental digital operations.
CO2	Design and simplify combinational logic circuits from Boolean expressions using Karnaugh Maps, and simulate them for correctness and efficiency.
CO3	Construct and analyze sequential circuits such as flip-flops, counters, and shift registers to demonstrate state behavior and timing sequences.
CO4	Develop arithmetic circuits and evaluate algorithmic performance (e.g., Booth's multiplication) using Hardware Description Languages (HDL).
CO5	Integrate combinational and sequential modules to create a simplified CPU architecture through collaborative mini-projects, enhancing teamwork, communication, and project management skills.

Course Content

Lab No.	Title	Description
1	Basic Logic Gates	Implement and verify truth tables of NOT, AND, OR, NAND, NOR, XOR, XNOR gates using ICs and simulation software.
2	Boolean Expression Simplification	Design logic circuits from Boolean expressions, simplify using Karnaugh Maps, and simulate the simplified circuit.
3	Combinational Circuit – Adders & Subtractors	Implement Half-Adder, Full-Adder, Half-Subtractor, and Full-Subtractor using logic gates and ICs.
4	Design of Code	Design and implement Binary to Gray, Gray to Binary,

	Converters	Binary to BCD, and BCD to Excess-3 converters.
5	Multiplexers and DE multiplexers	Design and verify 4:1, 8:1 MUX and 1:4, 1:8 DEMUX using logic gates and ICs or simulation tools.
6	Encoders and Decoders	Implement 8-to-3 encoder and 3-to-8 decoder using logic gates and analyze their truth tables.
7	Flip-Flops and Latches	Design and test SR, JK, D, T flip-flops using ICs or HDL; study race-around and master-slave configurations.
8	Synchronous and Asynchronous Counters	Design and simulate up/down counters (binary, mod-n, ring, Johnson) and study their timing behavior.
9	Shift Registers	Implement SISO, SIPO, PIPO, and PISO registers using flip-flops or HDL; demonstrate serial and parallel operations.
10	Arithmetic Circuits Using HDL	Model addition, subtraction, Booth's multiplication, restoring and non-restoring division using Verilog/VHDL.
11	Memory and Address Decoding	Simulate basic RAM/ROM using HDL and design address decoder circuits for memory mapping.
12	Mini Project / CPU Module Simulation	Group-based implementation of a simple CPU datapath (ALU + Register File + Control Unit) using HDL or simulation.

Tools and Resources:

Software: Logisim, Multisim, ModelSim, Xilinx Vivado / ISE, Quartus
Hardware Kits: Digital Trainer Kit, ICs (74xx series), LEDs, switches, Breadboards
Languages: Verilog/VHDL (optional for advanced simulation)

Textbooks:

3. Digital Logic and Computer Design by M. Morris Mano,Pearson Education, 1st Edition
4. Computer Organization and Architecture: Designing for Performance by William Stallings, Pearson Education, 10th Edition

Reference Books:

6. Digital Design by M. Morris Mano, Michael D. Ciletti, Pearson Education, 5th Edition
7. Computer Organization and Embedded Systems by Carl Hamacher, Zvonko Vranesic, Safwat Zaky,McGraw-Hill Education, 6th Edition
8. Computer Organization and Design: The Hardware/Software Interface by David A. Patterson, John L. Hennessy, Morgan Kaufmann Publishers, RISC-V Edition

9. Fundamentals of Logic Design by Charles H. Roth Jr., Larry L. Kinney,Cengage Learning, 7th Edition
1. Digital Fundamentals by*Thomas L. Floyd*, Pearson Education, 11th EditionFundamentals of Logic Design by Charles H. Roth Jr., Larry L. Kinney,Cengage Learning, 7th Edition
2. Digital Fundamentals by*Thomas L. Floyd*, Pearson Education, 11th Edition

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: ENGINEERING CHEMISTRY LAB
Paper Code: CH 291
Total Contact Hours: 24
Credit: 2
Prerequisites: 10+2

Course Objective

- Study the basic principles of pH meter and conductivity meter for different
- applications
- Analysis of water for its various parameters & its significance in industries
- Learn to synthesis Polymeric materials and drugs
- Study the various reactions in homogeneous and heterogeneous medium

Course Outcome

CO1: Able to operate different types of instruments for estimation of small quantities chemicals used in industries and scientific and technical fields.
CO2: Able to analyse and determine the composition and physical property of liquid and solid samples when working as an individual and also as a team member.
CO3: Able to analyse different parameters of water considering environmental issues.
CO4: Able to synthesize drug and sustainable polymer materials.
CO5: Capable to design innovative experiments applying the fundamentals of modern chemistry.

COURSE CONTENT

Any 10 experiments to be conducted preferably a combination of estimation, water quality analysis, instrumental analysis and synthesis

1. To determine strength of given sodium hydroxide solution by titrating against standard oxalic acid solution.
2. Estimation of amount of Fe^{2+} in Mohr's salt using permanganometry.
3. To determine the surface tension of a given liquid at room temperature using stalagmometer by drop number method.
4. To determine the viscosity of a given unknown liquid with respect to water at room temperature, by Ostwald's Viscometer.
5. Water quality analysis:
    i. Determination of total, permanent and temporary hardness of sample water by complexometric titration.

ii. Determination of Cl^- ion of the sample water by Argentometric method.
iii. Determination of alkalinity of the sample water.
iv. Determination of dissolved oxygen present in a given water sample.

6. Determination of the concentration of the electrolyte through pH measurement.
7. pH- metric titration for determination of strength of a given HCl solution against a standard NaOH solution.
8. Determination of cell constant and conductance of solutions.
9. Conductometric titration for determination of the strength of a given HCl solution by titration against a standard NaOH solution.
10. Determination of Partition Coefficient of acetic acid between two immiscible liquids.
11. Drug design and synthesis
12. Synthesis of polymers (Bakelite) for electrical devices and PCBs.
13. Synthesis of Silver Nanoparticles doped organic thin film for organic transistors.
14. Determination of R_F of any amino acid by thin layer chromatography.
15. Saponification /acid value of any oil.
16. Isolation of graphene from dead dry batteries

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

IDEA Lab Workshop

Course Code	**:**	**ME293**
Course Title	**:**	**IDEA Lab Workshop**
Number of Credits	**:**	**(L:0,T:0,P:3)**
Credit	**:**	**1.5**

Course Objectives:

1. To learn all the skills associated with the tools and inventory of the IDEA Lab.
2. To learn useful mechanical and electronic fabrication processes.
3. To learn the necessary skills to build useful and standalone systems/projects with enclosures.
4. To learn the necessary skills to create print and electronic documentation for the system/project.

Course Contents*:*

Module	Topics	
1	Electronic component familiarization. Understanding electronic system design flow. Schematic design and PCB layout and Gerber creation using Eagle CAD. Documentation using Doxygen, Google Docs, and Overleaf. Version control tools – GIT and GitHub. Basic 2D and 3D designing using CAD tools such as FreeCAD, SketchUp, PrusaSlicer, FlatCAM, Inkscape, OpenBSP, and VeriCUT.	**Introduction to basic hand tools:** Tape measure, combination square, Vernier calliper, hammers, fasteners, wrenches, pliers, saws, tube cutter, chisels, vice and clamps, tapping and threading, adhesives. **Introduction to power tools:** Power saws, band saw, jigsaw, angle grinder, belt sander, bench grinder, rotary tools, and various types of drill bits.

2	**Familiarization and use of basic measurement instruments:** DSO (Digital Storage Oscilloscope) including various triggering modes, DSO probes, DMM (Digital Multimeter), LCR bridge, signal and function generator, logic analyzer, and MSO (Mixed Signal Oscilloscope). Bench power supply (with 4-wire output). **Circuit prototyping using:** (a) Breadboard (b) Zero PCB (c) 'Manhattan' style (d) Custom PCB **PCB design and fabrication:** Single, double, and multilayer PCBs. Single and double-sided PCB prototype fabrication in the lab. **Soldering and assembly:** Soldering using soldering iron/station. Soldering using a temperature-controlled reflow oven. Automated circuit assembly and soldering using pick-and-place machines.	**Mechanical cutting processes:** 3-axis CNC routing, basic turning, milling, drilling, and grinding operations, laser cutting, laser engraving, etc. **Joining techniques:** Basic welding, brazing, and other joining techniques for assembly. **Concept:** Concept of "Lab aboard a Box."
3	**Electronic circuit building blocks:** Including common sensors. **Microcontroller and microprocessor platforms:** Arduino and Raspberry Pi – programming and use. **Core concepts:** Digital input and output, measuring time and events, PWM (Pulse Width Modulation), serial communication, analog input, and interrupts programming. **Power supply design:** Linear and switching types, wireless power supply, USB PD (Power Delivery), solar panels, battery types, and charging.	**3D Printing and Prototyping Technology:** 3D printing using FDM, SLS, and SLA. Basics of 3D scanning and point cloud data generation for reverse engineering. **Prototyping using subtractive processes:** 2D and 3D structures for prototype building using laser cutter and CNC routers. **Intellectual Property (IP) and Patents:** Basics of IP and patents; accessing and utilizing patent information in IDEA Lab.

4	Discussion and implementation of a mini project.
5	Documentation of the mini project (report and video).

Laboratory Activities:

Sl. No.	**List of Lab activities and experiments**
1.	Schematic and PCB layout design of a suitable circuit, fabrication and test of the circuit.
2.	Machining of 3D geometry on soft materials such as soft wood or modeling wax.
3.	3D scanning of computer mouse geometry surface. 3D printing of scanned geometry using FDM or SLA printer.
4.	2D profile cutting of press-fit box/casing in acrylic (3 or 6 mm thickness), cardboard, or MDF (2 mm) board using laser cutter and engraver.
5.	2D profile cutting on plywood/MDF (6–12 mm) for press-fit designs.
6.	Familiarity and use of welding equipment.
7.	Familiarity and use of normal and wood lathe.
8.	Embedded programming using Arduino and/or Raspberry Pi.
9.	Design and implementation of a capstone project involving embedded hardware, software, and a machined or 3D-printed enclosure.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

2nd Year 3rd Semester									
Sl.No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS301	Computer Architecture	3	0	0	3	3
2	ENGG	Major	CS302	Design and Analysis of Algorithms	3	0	0	3	3
3	ENGG	Major	CS303	Operating Systems	3	0	0	3	3
4	ENGG	Major	CS304	Advanced Artificial Intelligence	3	0	0	3	3
5	ENGG	Minor	EC(CS)301	Internet of Things	3	0	0	3	3
6	SCI	Minor	M(CS)301	Discrete Mathematics	3	0	0	3	3
B. PRACTICAL									
1	ENGG	Major	CS391	Computer Architecture Lab	0	0	3	3	1.5
2	ENGG	Major	CS392	Design and Analysis of Algorithms Lab	0	0	3	3	1.5
3	ENGG	Major	CS393	Operating Systems Lab	0	0	3	3	1.5
	ENGG	Major	CS394	Advanced Artificial Intelligence Lab	0	0	3	3	1.5
4	ENGG	Skill Enhancement Course	CS395	Python Programming Lab	0	1	3	4	2.5
5	ENGG	Minor	EC(CS)391	Internet of Things lab	0	0	3	3	1.5
Total of Theory, Practical								**36**	**28**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Computer Architecture
Course Code: CS301
Contact: 3:0:0
Total Contact Hours: 36L
Credits: 3

Prerequisite: Digital Electronics

Course Objectives:

1. To understand the fundamental concepts of computer organization and the architectural design of modern digital computers.
2. To explore how hardware and software interact at the instruction level to execute programs.
3. To analyze the performance of different processor architectures and memory systems and memory hierarchy.
4. To study the concepts of instruction set architecture (ISA), data path design, control unit design, and pipelining.
5. To provide insights into parallelism, including instruction-level, data-level, and processor-level parallel architectures.

Course Outcomes (COs): After attending the course students should be able to

CO1	Understand the basic structure and functionality of a CPU, including the Arithmetic Logic Unit (ALU), instruction formats, addressing modes. Analyze and apply fixed-point and floating-point arithmetic operations and IEEE 754 floating-point representation.
CO2	Apply the fundamental concepts of computer architecture, including stored program organization (Von Neumann and Harvard models), instruction set and differentiate between microprogrammed and hardwired control unit designs.
CO3	Analyze memory hierarchy and mapping techniques, and performance enhancements and evaluate virtual memory concepts and page replacement policies for efficient memory management.
CO4	Evaluate pipeline execution, hazards and the techniques and explore instruction-level parallelism and evaluate the functioning of array and vector processors in achieving parallel processing.
CO5	Create the fundamentals of multiprocessor architectures, Flynn’s taxonomy and analyze centralized and shared-memory systems and evaluate various interconnection networks in parallel computing systems.

Course Contents:

Module 1[8L]:

Introduction to CPU and concepts of ALU [2L], Instruction format and Instruction Cycle [1L], Addressing Modes [1L] Fixed- point multiplication -Booth's algorithm. [2L], Fixed-point division - Restoring and non-restoring algorithms [1L], Floating-point number representation-IEEE754 format and Floating-point arithmetic operation [1L].

Module 2 [7L]:
Introduction to basic computer architecture [1L], Stored Program Concepts: Von Neumann & Harvard Architecture [1L], RISC VS CISC [1L], Amdahl law [1L], Performance measurement parameters – MIPS, MFLOPS, SPEC ratings, CPI etc. [2L] Micro programmed and hardwired control unit [1L].

Module 3[8L]:
Introduction to memory-RAM and ROM [1L], Register transfer, memory transfer, Tri-state bus buffer, Memory Hierarchy: Secondary memory [1L], Main Memory [1L], Cache Memory [1L], Mapping Technique in cache memory: Direct, Full Associative and Set Associative [2L], Performance Implementation in Cache Memory [1L], Virtual memory Concepts [1L], page replacement policies [1L].

Module 4[9L]:
Pipelining: Basic concepts, instruction and arithmetic pipeline[2L], data hazards, control hazards and structural hazards, techniques for handling hazards[2L], Pipeline vs. Parallelism, Levels of parallelism [1L], Instruction- Level Parallelism: Basic Concepts, Techniques for Increasing ILP, Superscalar, Super Pipelined and VLIW Processor Architectures [2L], Array and Vector Processors[1L]

Module 5[4L]:
Multiprocessor architecture: taxonomy of parallel architectures; Flynn Classification [1L], Centralized and Shared- memory architecture: synchronization [1L], Interconnection Network (Omega, Baseline, Butterfly, Crossbar) [2L].

Text Books:
1. Hwang - Advanced Computer Architecture Parallelism Scalability Programmability, Tata McGraw- Hill Education Private Limited ISBN-13: 978-0-07-053070-6 ISBN-10:0-07-053070-X
2. Hwang & Briggs—Computer Architecture & Parallel Processing, TMH

Reference Books:
1. Patterson D.A. and Hennessy, J.L.―Computerarchitectureaquantitativeapproach‖,2nd ed. Morgan Kaufman,1996
2. Hayes J. P., ―Computer Architecture & Organization, McGrawHill

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

CO-PO Mapping

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

CO-PSO Mapping

COs	PSO1	PSO2	PSO3
CO1	3	3	3
CO2	3	2	3
CO3	2	3	3
CO4	3	3	2
CO5	3	3	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Design & Analysis of Algorithms
Course Code: CS302
Contact: 3:0:0
Credit Point: 3
Total Contact Hours: 36L

Prerequisites: To know data-structure and basic programming ability

Course Objectives:
To introduce the fundamental stages of algorithm development and provide students with the ability to analyze the time and space complexity using appropriate asymptotic notations.

To equip students with a variety of algorithm design techniques—such as brute force, divide-and-conquer, greedy algorithms, dynamic programming, and backtracking—to solve classical and real-world computational problems.

To familiarize students with advanced algorithmic strategies for string matching, graph traversal, shortest path problems, and network flow, enabling them to model and solve problems efficiently.

To build a strong foundation in computational complexity, NP-completeness, and problem reducibility, while also introducing approximation and randomized algorithms as approaches to handle intractable problems.

Course Outcomes (COs): After attending the course students should be able to

CO1: To understand and illustrate the concepts of time and space complexity, worst case, average case and best-case complexities and the asymptotic notation.
CO2: To analyze and apply the design principles and concepts to various basic algorithm design viz. dynamic programming, greedy methods etc.
CO3: To understand and analyze various string matching and graph algorithms.
CO4: To understand, illustrate and analyze the different complexity classes
CO5: To discuss, implement and analyze, verify the efficiency of the randomized and approximation algorithms.

Course Content:
Module-1 [6L]
Algorithm Development & Complexity Analysis: [4L] Stages of algorithm development for solving a problem: Describing the problem, identifying a suitable technique, Design of an algorithm, Time and Space Complexity, Different Asymptotic notations – their mathematical significance. Solving Recurrences: Substitution Method, Recurrence Tree Method, Master Theorem (Statement Only).

Module-2 [12L]
Algorithm Design Techniques Brute force techniques – Traveling Salesman Problem, Divide and Conquer - Matrix multiplication: Strassen algorithm, Greedy techniques - Fractional Knapsack problem, Job Sequencing with Deadline, Graph Coloring, Finding Minimum Cost

Spanning Tree using Prim's and Kruskal's algorithm, Huffman coding, Dynamic programming - O/1 Knapsack problem, Matrix chain multiplication, Travelling Salesman Problem, Backtracking-N-Queens Problem, Subset Sum Problem

Module-3 [3L]
String matching problem: Different techniques – Naive algorithm, Knuth, Morris, Pratt (KMP) algorithm with their complexities.

Module-4 [5L]
Graph Algorithms Single Source Shortest Path - Dijkstra Algorithm, Bellman-ford Algorithm, All pair shortest path – Floyd - Warshall Algorithm. Network Flows, Maximum Flows – Ford-Fulkerson Algorithm, Push Re-label Algorithm.

Module-5 [10L]
Complexity Classes: The Class P, The Class NP, Reducibility and NP-completeness – SAT (without proof), 3-SAT, Vertex Cover, Independent Set, Maximum Clique. Clique Decision problem and reduction of NP completeness.
Overview of Approximation and Randomized Algorithms, Recent Trends

Text book:
1. "Introduction to Algorithms" by Cormen, Leiserson, Rivest, Stein, The MIT Press, Cambridge, Massachusetts.
2. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman, Pearson Education (Originally published by Addison-Wesley), New Delhi.
3. "Algorithm Design" by Kleinberg and Tardos, Pearson Education, New Delhi.

Reference Books:
1. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House, New Delhi
2. Design Analysis and Algorithms‖ by Hari Mohan Pandey, niversity Science Press (An imprint of Laxmi Publications), New Delhi

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO1	PSO2	PSO3
CO 1	2	2	2	2	-	-	-	-	-	-	-	3	3	3
CO 2	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO 3	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO 4	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO 5	3	3	3	3	-	-	-	-	-	-	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Operating Systems
Course Code: CS303
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites:
1. Computer organization
2. Computer Architecture
3. Data Structures
4. Algorithms & Programming Concept

Course Objective(s):

- To understand the fundamental concepts, structure, and functionalities of operating systems, including process, memory, file, and I/O management.
- To explore process synchronization, scheduling, inter-process communication, and deadlock handling for efficient resource utilization and system performance.
- To analyse and apply various memory allocation, disk scheduling, and file system techniques used in modern operating systems.

Course Outcome(s):

After completion of the course students will be able

CO1: To understand and illustrate the structure, functions, and types of operating systems and their role in resource management.

CO2: To analyze and apply process management techniques, including scheduling, synchronization, and inter-process communication.

CO3: To evaluate memory management strategies such as paging, segmentation, and virtual memory.

CO4: To implement techniques for handling deadlocks and ensuring system reliability

CO5: To discuss the organization and management of I/O systems, disk scheduling algorithms, and file systems.

Course Content:
Module – I [4L]
Concepts of Operating System, Evolution of Operating System. Types of Operating System, Structural overview, Operating system services.

Module II [8L]
Processes: Concept of processes, transition of process states, Process Control Block, process scheduling, co-operating processes, independent process, suspended process, Interaction between processes and OS.
Threads: overview, benefits of threads, user and kernel level threads.
CPU scheduling: Scheduling criteria, preemptive & non-preemptive scheduling, scheduling algorithms (FCFS, SJF, SRTF, RR, priority, multilevel queue, multilevel feedback queue scheduling), Real Time scheduling: RM and EDF.
Case Study: Inter-process communication: Message passing. Thread models.

Module III [8L]
Inter-process Communication: background, critical section problem, synchronization hardware, Peterson's Solution, The Producer Consumer Problem, Semaphores.
Classical Problems of synchronization: Reader's & Writer Problem, Dinning Philosopher Problem, Monitors.

Module IV [4L]
Deadlocks: Definition, Necessary and sufficient condition for deadlock, methods for handling deadlocks: deadlock prevention, deadlock avoidance: Banker's algorithm, deadlock detection, recovery from deadlock.

Module V [6L]
Background, logical vs. physical address space, swapping, contiguous memory allocation, paging, Segmentation, TLB. Virtual Memory: background, demand paging, page replacement algorithms (FCFS, LRU, Optimal), thrashing, Working set model.

Module VI [6L]
I/O Hardware: I/O devices, polling, interrupts, DMA, caching, buffering, blocking-non blocking I/O.
Disk Management: Disk structure, disk scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK etc), disk reliability, disk formatting, boot block, bad blocks.
File: File concept, access methods, directory structure, file system structure, UNIX file structure, allocation methods (contiguous, linked, indexed), free-space management (bit vector).

Text Book:
1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts.
2. Operating systems: concepts and design by Milan Milenkovič- Mc. Graw-Hill Publication

Reference Book:
1. Dietel H. N., ―An Introduction to Operating Systems‖, Addison Wesley.
2. Andrew Tanenbaum, Modern Operating Systems, Prentice Hall.
3. William Stallings, Operating Systems, Prentice Hall.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	-	-	-	-	-	-	-	2	2	3
CO2	3	3	3	3	-	-	-	-	-	-	-	2	3	3
CO3	3	3	3	2	-	-	-	-	-	-	-	2	3	2
CO4	3	3	2	3	-	-	-	-	-	-	-	3	2	3
CO5	3	3	3	2	-	-	-	-	-	-	-	2	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Advanced Artificial Intelligence
Course Code: CS304
Contact:- 3:0:0
Total Contact Hours: 36L
Credits: 3

Prerequisite:
1. Basics of Design and Analysis of Algorithm
2. A solid background in mathematics, including probability.

Course Objective(s)

- To learn the overview of artificial intelligence principles and approaches.
- To develop a basic understanding of the building blocks of AI as presented in terms of intelligent agents.
- This course also covers fundamental areas of Local Search Algorithms, Adversarial
- Searching and Neural Networks.

Course Outcome(s):
On completion of the course students will be able
CO1: To understand the concepts of Artificial intelligence, Intelligent agents and Learning
CO2: To Express knowledge of the world using logic and infer new facts from that Knowledge.
CO3: To implement intelligent algorithms for constraint satisfaction problems and game playing.
CO4: To apply Artificial intelligence Techniques in different fields like Natural Language Processing and Expert Systems.
CO5: To analyze working knowledge in Python in order to write simple Python programs and
explore more sophisticated Python code on their own.

Module 1: Basics of AI [6L]
Introduction [2]
Overview of Artificial intelligence- Problems of AI, AI technique, Tic - Tac - Toe problem.
Intelligent Agents [2]
Agents & environment, nature of environment, structure of agents, goal-based agents, utility based agents, learning agents.
Learning [2]
Forms of learning, inductive learning, learning decision trees, explanation based learning, learning using relevance information, neural net learning & genetic learning.

Module 2: Different types of searching algorithms [14L]
Problem Solving [3]
Problems, Problem Space & search: Defining the problem as state space search, production

system, constraint satisfaction problems, issues in the design of search programs.
Search techniques [4]
Solving problems by searching: Problem solving agents, searching for solutions; uniform search strategies: breadth first search, depth first search, depth limited search, bidirectional search, comparing uniform search strategies.
.
Heuristic search strategies [4]
Greedy best-first search, A* search, memory bounded heuristic search: local search algorithms & optimization problems: Hill climbing search, simulated annealing search, local beam search, genetic algorithms; constraint satisfaction problems, local search for constraint satisfaction problems.
Adversarial search [3]
Games, optimal decisions & strategies in games, the minimax search procedure, alpha-beta pruning, additional refinements, iterative deepening.

Module 3: Knowledge & Reasoning [12L]
Knowledge & Reasoning: Knowledge representation issues, representation & mapping, approaches to knowledge representation, issues in knowledge representation.
Using predicate logic: Representing simple fact in logic, representing instant & ISA relationship, computable functions & predicates, resolution, natural deduction.
Representing knowledge using rules: Procedural verses declarative knowledge, logic programming, forward verses backward reasoning, matching, control knowledge.
Probabilistic reasoning: Representing knowledge in an uncertain domain, the semantics of Bayesian networks, Dumpster Shafer theory, Fuzzy sets, and fuzzy logics.

Module 4: Different fields of AI [4L]
Natural Language Processing: Introduction, Syntactic processing, semantic analysis, discourse, and pragmatic processing.
Expert Systems: Representing and using domain knowledge, expert system shells, and knowledge acquisition. Basic knowledge of programming language like Python

Text books:
1. Artificial Intelligence, Ritch & Knight, TMH
2. Artificial Intelligence, A Modern Approach, Stuart Russel, Peter Norvig, Pearson

Recommended books:
3. Introduction to Artificial Intelligence & Expert Systems, Patterson, PHI
4. Poole, Computational Intelligence, OUP
5. Expert Systems, Giarranto, VIKAS

COs	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PSO1**	**PSO2**	**PSO3**
CO 1	2	2	2	2	-	-	-	-	-	-	-	3	3	3

CO 2	3	3	2	3	-	-	-	-	-	-	-	3	3	3
CO 3	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO 4	3	2	3	3	-	-	-	-	-	-	-	3	3	3
CO 5	2	3	3	3	-	-	-	-	-	-	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Internet of Things
Course Code: EC(CS)301
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisite:

1. Fundamental knowledge in computer networking.

2. Basic knowledge of Microcontroller fundamentals.

Course Objective(s):

Students will understand the concepts of Internet of Things and be able to build IoT applications.

Course Outcome(s):

On completion of the course students will be able to

EC(CS)301.1: Understand and differentiate the concepts of Internet of Things and Internet.

EC(CS)301.2: Identify appropriate MAC protocols and routing protocols while solving a problem.

EC(CS)301.3: Analyze and compare the basic protocols in wireless sensor network and IoT.

EC(CS)301.4: Solve different real-life problems in different domains based upon the concept of IoT and sensor network.

EC(CS)301.5: Implement basic IoT applications on embedded platforms.

Course Content:

Module I: [7L]
Fundamentals of IoT: The Internet of Things, Time for Convergence, Towards the IoT Universe, Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Applications, Future Internet Technologies, Infrastructure, Networks and Communication, Design challenges, Development challenges, Security challenges, Other challenges.

Module II: [6L]
Wireless Sensor Network, Network & Communication aspects, Wireless medium access issues, MAC protocol, routing protocols, Sensor deployment & Node discovery, Data aggregation & dissemination.

Module III: [7L]
IoT and M2M A Basic Perspective– Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

Module IV: [7L]
IoT Architecture Introduction, Architecture Reference Model- Introduction, Reference Model, and architecture, IoT reference Model, IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.

Module V: [5L]
IoT Applications for Value Creations Introduction to Arduino and Raspberry Pi, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Introduction, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications, Four Aspects in your Business to Master IoT, Value Creation from Big Data and Serialization, IoT in healthcare, Value for Industry, smart home Management.

Module VI: [4L]
Internet of Things Privacy, Security and Governance Introduction, Overview of Governance, Privacy and Security Issues, Trust in IoT-Data Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in smart cities, Security.

Text books:
1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014.
2. Francis da Costa, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013.

Reference books:
1. Cuno Pfister, Getting Started with the Internet of Things, O‟Reilly Media, 2011, ISBN: 978- 1-4493-9357-1
2. Waltenegus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice"

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	2	-	-	-	-	-	-	1		
CO2	3	3	3	3	2	-	-	-	-	-	-	1		
CO3	3	3	3	3	2	-	1	-	-	-	-	1	2	
CO4	3	3	3	3	3	2	2	-	-	-	-	3		
CO5	3	3	3	3	3	2	2	-	2	2	-	3		

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: DISCRETE MATHEMATICS
Course Code: M(CS)301
Contact: (L: T: P): 3: 0: 0
Total Contact Hours: 36
Credit: 3

Prerequisites:
The students to whom this course will be offered should have a fundamental understanding of (10+2) standard set theory, algebra, and logic, along with the ability to follow formal mathematical notation and basic proof techniques.

Course Objective(s):
The objective of the course is to make the students able to –

1. Develop a conceptual understanding of set theory, relations, lattices, combinatorics, and propositional logic to model discrete mathematical systems.
2. Gain proficiency in applying number theory and algebraic structures—such as groups, rings, and fields—for logical reasoning and problem-solving in computing contexts.
3. Build the skill to tackle problems involving graphs, trees, and recurrence relations using algorithmic approaches and generating functions.

Course Outcomes (COs):

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

CO1	Apply fundamental concepts of set theory, relations, lattices, and combinatorics to model structured and logical problems in computing.
CO2	Apply number theoretic techniques and algebraic structures such as groups and rings in the design and analysis of secure and efficient computational systems.
CO3	Analyze logical propositions using propositional logic and truth tables to draw valid conclusions and verify the consistency of logical systems.
CO4	Analyze the structural properties and relationships within computational and engineering problems using graph theory and tree-based algorithms.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CO1	3	2	-	-	-	-	-	-	-	-	1			
CO2	3	2	-	-	-	-	-	-	-	-	1			
CO3	3	3	1	1	-	-	-	-	-	-	2			
CO4	3	3	1	1	-	-	-	-	-	-	2			
M(CS)301	**3**	**2.5**	**1**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**1.5**			

Weightage Values: Strongly mapped: ‘3’, Moderately mapped: ‘2’, Weakly mapped: ‘1’, Not mapped: ’-‘.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content:

Module-I: Set Theory [11L]

Posets & Lattices: [6L]

Relation: Types of Relations, Properties of Binary Relation, Equivalence Relation, Partial Ordering Relation and Posets, Lattices.

Combinatorics: [2L]

Principle of Inclusion Exclusion, Pigeon Hole Principle.

Generating Functions and Recurrence Relations: [3L]

Generating functions, Recurrence relations: Formulation of different counting problems in terms of recurrence relations, Solution of recurrence relations with constant coefficients by Generating functions method.

Module-II: Propositional Logic[5L]

Basics of Boolean Logic, Idea of Propositional Logic, well-formed formula, Logical Connectives, Truth tables, Tautology, Contradiction, Algebra of proposition, Logical Equivalence, Normal Forms: Disjunctive Normal Forms (DNF) and Conjunctive Normal Forms (CNF).

Module-III: Number Theory [4L]

Well-Ordering Principle, Divisibility theory and properties of Divisibility, Fundamental theorem of Arithmetic, Prime and Composite Numbers, Greatest Common Divisor and Euclidean Algorithm, Congruence, Residue Classes.

Module-IV: Algebraic Structures [8L]

Concepts of Groups, Subgroups and Order, Cyclic Groups, Cosets, Lagrange's theorem,Normal Subgroups, Permutation Groups and Symmetric Groups, Definition of Ring and Field.

Module-V: Graph Theory [8L]

Graph theory, Theorems (statement only), Digraphs, Weighted Graph, Walk, Path, Circuit, Connected and Disconnected Graph, Bipartite Graph, Complement of a Graph, Regular Graph, Complete Graph, Adjacency and Incidence matrices of a graph (digraph), Dijkstra's algorithm.

Tree, Binary Tree, Theorems on Tree (statement only), Spanning Tree, Minimal Spanning Tree, Kruskal's Algorithm, Prim's Algorithm.

Text Books:

1. Graph Theory with Applications to Engineering and Computer Science, Deo, N., Prentice Hall.

2. Higher Algebra: Abstract and Linear, Mapa, S. K., Levant, 2011.
3. Discrete Mathematics, Chakraborty, S. K. and Sarkar,B. K., OXFORD University Press.
4. Discrete Mathematics and its Applications, Rosen, K. H., Tata McGraw – Hill.

Reference Books:

1. Higher Engineering Mathematics, Grewal, B. S., Khanna Pub.
2. Advanced Engineering Mathematics, Kreyzig, E., John Wiley and Sons.
3. Discrete Mathematics, Sharma, J.K., Macmillan.
4. Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition, Liu, C. L. and Mohapatra, D. P., Tata McGraw – Hill.
5. Discrete Mathematical Structure and Its Application to Computer Science, TMG Edition, Tremblay, J. P. and Manohar, R., TataMcgraw-Hill.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Computer Architecture Lab
Course Code: CS391
Contact: 0:0:3
Credits: 1.5

Prerequisites:
Knowledge of designing different circuits in Computer Organization Lab

Course Objectives
To familiarize students with the fundamentals of digital logic gates and enable them to design, simulate, and verify combinational logic circuits such as adders, subtractors, multiplexers, and encoders.

To develop the ability to implement and analyze sequential circuits including flip-flops, shift registers, and memory units through simulation tools.

To provide hands-on experience in the design and functioning of an Arithmetic Logic Unit (ALU) and basic RAM architecture, fostering a deeper understanding of digital system components.

To encourage creativity and innovation by engaging students in designing and simulating customized or application-specific digital circuits beyond standard experiments.

Course Outcomes (COs): After attending the course students should be able to

CO1	Illustrate and use proper syntax in appropriate platforms for developing programs to solve problems related to Mathematics and Engineering fields leading to lifelong learning.
CO2	Apply the knowledge of algorithms in the computational area to efficient programming codes to design the CO3 problem using modern tools for solving complex engineering problems.
CO3	Outline different types of digital electronic circuits such as adder, subtract or, encoder decoder, multiplexer, demultiplexer, flip-flops, register, counter using various mapping and modern tools to prepare the most simplified circuit and optimize using various mapping and mathematical methods for solving the problem as a professional engineering practice as a team.
CO4	Apply the knowledge of digital electronic circuits to design memory and ALU and analyze the same to solve engineering-related computational problems as a team.
CO5	Interpret the result of the experiments, prepare laboratory reports based on observed output and analyze it to validate professional ethics and responsibilities and norms of the engineering practice.

List of Experiment:

1. Implement different types of Basic gates and simulate for truth table verification.
2. Implement half adder circuit and simulate for truth table verification.
3. Implement full adder circuit and simulate for truth table verification.
4. Implement half subtractor circuit and simulate for truth table verification.
5. Implement a full subtractor circuit and simulate for truth table verification.
6. Implement Multiplexer, De-Multiplexer circuit and simulate for truth table verification.
7. Implement Encoder, Decoder circuit and simulate for truth table verification.
8. Implement different types of flip flop and simulate for truth table verification.
9. Implement different types of parallel circuits (SISO, SIPO, PISO, PIPO) and simulate the result.
10. Implement ALU and simulate the result.
11. Implement a RAM chip and simulate the result.
12. Innovative Experiments.

CO-PO Mapping

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

CO-PSO Mapping

COs	PSO1	PSO2	PSO3
CO1	3	3	3
CO2	3	3	3
CO3	3	3	3
CO4	3	3	3
CO5	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Design & Analysis of Algorithm Lab
Course Code: CS392
Contact: 0:0:3
Credit Point: 1.5

Prerequisite: Programming knowledge

Course Objectives
1. To provide hands-on experience in implementing algorithmic paradigms such as Divide and Conquer, Greedy, Dynamic Programming, Backtracking, and Branch & Bound to solve computational problems.

2. To enable students to implement and analyze the performance of classical algorithms for real-world problems including string matching, shortest path determination, and combinatorial optimization.

3. To strengthen students' ability to convert algorithmic concepts into code using suitable data structures and assess their time and space complexity.

4. To encourage problem-solving skills by applying algorithmic strategies to real-life and emerging trend-based problems through simulation and practical implementation.

Course Outcomes (COs): After attending the course students should be able to
CO1: To identify and prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains.
CO2: To understand and illustrate methods for analyzing the efficiency and correctness of algorithms (such as exchange arguments, recurrence, induction, and average case analysis)
CO3: To analyze and design algorithms using the dynamic programming, greedy method, Backtracking, Branch and Bound strategy, and recite algorithms that employ this strategy.
CO4: To understand, compare, contrast, and choose appropriate implementation of the algorithmic design techniques to present an algorithm that solves a given problem.
CO5: To Identify and analyze criteria and specifications appropriate to new problems.

Course Content:

A. Implementation of various Divide & Conquer Methods; viz. Matrix Multiplication.
B. Implementations of various Dynamic Programming Methods, viz. Matrix Chain Multiplication Method, Travelling Salesman Problem etc.
C. Implementations of various Branch & Bound Techniques, viz.
D. Implementations of various Backtracking Methods, viz. n-Queen Problem.
E. Implementations of Greedy Method, viz. Fractional Knapsack Problem, Job Sequencing Problem etc.
F. Implementations of String-matching Algorithm viz. Naïve Algorithm, String Matching with KMP algorithm
G. Implementations of Various Graph Algorithms, viz. Dijkstra 's Algorithm, Floyd

Algorithm etc.
H. Implementation of some Real-Life Trendy Problems

Text book:
1. "Introduction to Algorithms" by Cormen, Leiserson, Rivest, Stein, The MIT Press, Cambridge, Massachusetts.
2. "Algorithm Design" by Kleinberg and Tardos, Pearson Education, New Delhi.

Reference Books:
1. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House, New Delhi
2. Design Analysis and Algorithms‖ by Hari Mohan Pandey, niversity Science Press (An imprint of Laxmi Publications), New Delhi

CO–PO Mapping:

COs	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	2	2	2	2	-	-	-	-	-	-	-	3	3	3
CO2	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO3	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO4	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO5	3	3	3	3	-	-	-	-	-	-	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Operating Systems Lab
Course Code: CS393
No. of Lectures: 36

Prerequisites:

1. Computer organization
2. Computer Architecture
3. Data Structures
4. Algorithms & Programming Concept

Course Objectives

1. To introduce essential Linux commands for effective file handling, process monitoring, and basic system operations, enabling students to navigate and manage a Unix/Linux environment proficiently.

2. To develop students' ability to write and execute shell scripts using variables, control structures, and functions for automating tasks and performing system-level operations.

3. To provide hands-on experience in managing processes, including creation, duplication, and replacement of process images, fostering an understanding of process control mechanisms.

4. To enable students to write concurrent programs using inter-process communication (IPC) mechanisms like semaphores and POSIX threads, ensuring effective synchronization and multithreading.

Course Outcome(s):

After completion of the course students will be able to

CO1: Use essential Linux commands for file handling, process management, and system operations effectively.
CO2: Execute shell scripts using variables, control structures, and built-in commands.
CO3: Manipulate processes, including creation, duplication, and replacement of process images.
CO4: Implement process synchronization using System V semaphores for managing concurrent access.
CO5: Create and manage POSIX threads to build multithreaded applications using appropriate thread-handling functions.

Course Content:

1.Essential Linux Commands[9P]: Commands for files and directories cd, cp, mv, rm, mkdir, more, less, creating and viewing files, using cat, file comparisons, View files, kill, ps, who, sleep, grep, fgrep, find, sort, cal, banner, touch, file related commands – ws, sat, cut, grep etc. Mathematical commands –expr, factor, units, Pipes (use functions pipe, popen, pclose), named Pipes (FIFOs, accessing FIFO)

2.Shell Programming [9P]: Creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, and commands).
3.Process [6P]: Starting new process, replacing a process image, duplicating a process image.
4.Semaphore [6P]: Programming with semaphores (use functions semget, semop, semaphore_p, semaphore_v).
5.POSIX Threads[6P]: Programming with pthread functions (viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel).

Text Books:

1. Yashavant P. Kanetkar, UNIX Shell Programming, 1st edition, BPB Publications
2. W. Richard Stevens, UNIX Network Programming, 2nd edition, Prentice Hall
3. Sumitabha Das, UNIX concepts and applications, 4th edition, TATA McGRAW HILL

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	-	-	-	-	3	-	-	2	2	3
CO2	3	3	3	3	-	-	-	-	3	-	-	2	3	3
CO3	3	3	3	2	-	-	-	-	2	-	-	2	3	3
CO4	3	3	2	3	-	-	-	-	3	-	-	3	2	3
CO5	3	3	3	2	-	-	-	-	3	-	-	2	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Artificial Intelligence Lab
Course Code: CS394
Contact (Periods/Week): 0:0:3
Credit Point: 2.5
No. of Lectures: 36

Course Name:
Objective(s):

- To learn the fundamentals of Python Programming.
- To impart adequate knowledge on the need of Python programming languages and problem-solving techniques.

Course Outcome(s):
On completion of the course students will be able to
CO1: To understand the concept of simple programming using Python.
CO2: To understand the concept of AI based programs using Python.
CO3: To apply logic based techniques in various real life problem domains.
CO4: To analyze logic based techniques in various real life problem domains.
.
List of Experiments:
Unit 1 – Python Basics for AI
Data types, loops, functions, file handling. Introduction to NumPy and Pandas.
Data Preprocessing: Importing datasets. Handling missing values, normalization, encoding categorical variables.

Unit 2 – Problem-Solving and Search Algorithms
State Space Search: Implement BFS (Breadth-First Search) and DFS (Depth-First Search). Pathfinding in a maze using Python. Informed Search: Implement Best-First Search and A* Algorithm. Apply to shortest path problems.

Unit 3 – Knowledge Representation and Reasoning
Propositional Logic: Implement truth tables and logical inference. Constraint Satisfaction Problems (CSP): N-Queens Problem using Backtracking.

Unit 4 – Machine Learning with Python
Supervised Learning: Implement Linear Regression and Logistic Regression using Scikit-learn. Classification using Decision Trees and K-NN.
Unsupervised Learning: K-Means clustering on a dataset. Dimensionality reduction using PCA.

Unit 5 – Natural Language Processing (NLP)
Text Processing with NLTK / SpaCy: Tokenization, stopword removal, stemming, lemmatization. Sentiment analysis on a dataset.

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	2	-	-	-	-	-	-	-	3	2	2
CO 2	2	3	2	2	-	-	-	-	-	-	-	1	3	3
CO 3	2	3	3	2	-	-	-	-	-	-	-	1	2	3
CO 4	2	3	2	2	-	-	-	-	-	-	-	2	2	2
CO 5	3	3	3	3	-	-	-	-	-	-	-	2	3	2

Reference book:

1. Learning Python, Mark Lutz, O'REILY.
2. Python Programming - Using Problem Solving Approach, Reema Thareja, OXFORD UNIVERSITY PRESS.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Python Programming Lab
Course Code: CS395
Contact (Periods/Week): 0:1:3
Credit Point: 2.5
No. of Lectures: 36

Prerequisites: Knowledge of Mathematics and basic concepts of Programming

Course Objectives

1.To introduce the fundamentals of Python programming, including variables, data types, operators, control structures, and functions, enabling students to develop basic programs with proper syntax and logic.

2. To provide hands-on experience in working with Python's built-in data structures such as strings, lists, sets, and dictionaries, fostering the ability to manipulate and organize data effectively.

3. To develop proficiency in file handling operations, modular programming through packages and modules, and dynamic data processing using libraries like NumPy and Pandas.

4. To equip students with the skills needed to perform efficient numerical and tabular data operations using NumPy and Pandas for real-world data analysis and scientific computing.

Course Outcome(s):
After completion of the course students will be able to
CO1: Understand basic of Python Programming Language
CO2: Understand the use of Conditional statement and Loops
CO3: Learn functions in python and represent collection type data using List and Dictionary
CO4: Read and write data from & to files in Python
CO5: Understand Numpy array and numerical operations on Numpy array.

Course Contents:
1. Basics of Python: Python Installation, Python variables, data types and Operator.
2. Loops: While and For loops, Python Syntax, Colon & Indentation, Conditional Statements: if,
elseif and else.
3. Functions: Defining Functions in python; passing arguments.
4. String: Python Programming to explore string functions
5. Lists: Python programs using Lists; understand the use of List methods, Slicing on List.
6. Sets: Working with Sets; Write programs to show different set operations.
7. Dictionary: Demonstrate the use of Dictionaries
8. File handling: Reading & Writing data from a file, Redirecting output streams to files.
9. Basics of Module & Package: Creation, Updation, Deletion.
10. Numpy: Numerical operations using NumPy array; slicing numpy array; stacking numpy

arrays; Write programs to show different numerical operations on NumPy array;
11. Pandas: Numerical operations using Pandas array; slicing Pandas array; stacking Pandas arrays; Write programs to show different numerical operations on Pandas;

Text book:

1. Zed A. Shaw, ADDISON-WESLEY, "Learn Python The Hard Way".
2. Introduction to Computing & Problem Solving With PYTHON, Jeeva Jose, Khanna Publishing House.
3. Introduction To Python Programming, Venkatesh, Nagaraju Y, Khanna Publishing House.

Reference book:

3. Learning Python, Mark Lutz, O'REILY.
4. Programming In Python, Dr. Pooja Sharma, BPB.
5. Python Programming - Using Problem Solving Approach, Reema Thareja, OXFORD UNIVERSITY PRESS.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO 1	2	-	-	2	-	-	-	-	-	-	-	3	2	2
CO 2	2	3	-	2	-	-	-	-	-	-	-	1	3	3
CO 3	2	3	3	2	-	-	-	-	-	-	-	1	2	3
CO 4	2	3	2	2	-	-	-	-	-	-	-	2	2	2
CO 5	3	3	3	3	-	-	-	-	-	-	-	2	3	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Internet of Things Lab
Course Code: EC(CS)391
Contact (Periods/Week): 0:0:3
Credit Point: 1.5
No. of Lectures: 36

Prerequisite:

1. Fundamental knowledge in computer networking.

2. Basic knowledge of Microcontroller fundamentals.

Course Objective(s):

Students will understand the concepts of Internet of Things and be able to build IoT applications.

Course Objectives

1. To introduce the foundational concepts of the Internet of Things (IoT) and develop basic skills in programming microcontrollers (Arduino) for sensor data acquisition and control applications.

2. To enable students to interface a variety of sensors and actuators, understand signal handling, and control physical devices based on real-time input.

3. To familiarize students with wireless communication modules and IoT protocols, and to develop the ability to transmit data to cloud platforms using modules like NodeMCU and ESP8266.

4. To promote the integration of multiple IoT components for building real-world applications in smart homes, health monitoring, and agriculture, and to provide exposure to cloud integration and data visualization.

5. To encourage project-based learning by implementing a complete IoT system, including hardware/software integration, debugging, and documentation of the design and outcomes.

Course Outcome(s):

On completion of the course students will be able to

EC(CS)391.1: Understand and differentiate the concepts of Internet of Things and Internet.

EC(CS)391.2: Identify appropriate MAC protocols and routing protocols while solving a problem.

EC(CS)391.3: Analyze and compare the basic protocols in wireless sensor network and IoT.

EC(CS)391.4: Solve different real-life problems in different domains based upon the concept of IoT and sensor network

EC(CS)391.5: Implement basic IoT applications on embedded platforms.

Course Content:

Module I: Introduction to IoT and Arduino Programming [6]
Objective: Understand the basics of IoT and learn how to program microcontrollers (Arduino).

- Overview of IoT: Architecture, applications, and components
- Introduction to Arduino IDE and boards (UNO/Nano)
- Basic digital and analog I/O operations
- Blinking LED, serial communication
- Reading data from sensors (e.g., temperature, light)

Experiments:

- LED blink program
- Serial monitor for sensor data display
- Temperature reading using LM35 sensor

Module II: Sensor Interfacing and Data Acquisition [6]
Objective: Learn how to interface different sensors and collect data.

- Interfacing analog and digital sensors
- Using potentiometers, IR sensors, and motion detectors
- Data acquisition techniques
- Noise filtering basics

Experiments:

- Motion detection using PIR sensor
- Distance measurement using ultrasonic sensor
- Analog sensor graph plotting (e.g., LDR)

Module III: Actuators and Output Control [6]
Objective: Control actuators based on sensor input.

- Types of actuators: motors, relays, buzzers
- Interfacing DC motors and servos

- Relay module interfacing (for home automation)

Experiments:

- Fan or light control using temperature sensor
- Servo motor control using potentiometer
- Home automation demo with relay

Module IV: Communication Modules and IoT Protocols [6]
Objective: Enable devices to communicate wirelessly.

- Overview of IoT protocols: MQTT, HTTP, CoAP
- ESP8266 / NodeMCU basics
- Sending data to cloud platforms (ThingSpeak, Blynk)
- Introduction to Wi-Fi, Bluetooth, and LoRa modules

Experiments:

- Wi-Fi-based temperature logger using NodeMCU + DHT11 + ThingSpeak
- Remote LED control using Blynk app
- Bluetooth-based home appliance control

Module V: IoT Applications and Integration [6]
Objective: Develop mini-projects integrating multiple IoT components.

- Smart home use cases
- Health monitoring systems
- Smart agriculture/irrigation system
- Integration with cloud dashboards
- Basics of data visualization

Experiments:

- Smart irrigation system (moisture sensor + relay + pump)
- IoT-based health monitoring (heartbeat + temperature sensors)
- Fire detection and alert system (flame sensor + buzzer + SMS)

Module VI: Final Project and Report Preparation [6]
Objective: Implement a full IoT-based system and document it.

Experiments:

- Project selection and design
- Integration of hardware and software components
- Testing, troubleshooting, and refinement
- Project demonstration and viva
- Lab report writing

Text books:

1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014.

2. Francis da Costa, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013.

Reference books:

1. Cuno Pfister, Getting Started with the Internet of Things, O’Reilly Media, 2011, ISBN: 978- 1-4493-9357-1

2. Waltenegus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice"

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	3	2	2	-	-	-	-	-	-	1		
CO2	2	2	3	2	2	-	-	-	-	-	-	1		
CO3	2	2	2	2	2	1	-	-	-	-	-	1	2	
CO4	3	2	2	2	2	2	2	-	-	-	-	2		
CO5	2	3	2	2	3	2	2	-	2	2	-	2		

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

2nd Year 4th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS401	Database Management Systems	3	0	0	3	3
2	ENGG	Major	CS402	Computer Networks	3	0	0	3	3
3	ENGG	Major	CS403	Machine Learning	3	0	0	3	3
4	ENGG	Major	CS404	Formal Language and Automata Theory	3	0	0	3	3
5	SCI	Minor	M(CS)401	Probability and Statistics	3	0	0	3	3
B. PRACTICAL									
1	ENGG	Major	CS491	Database Management Systems Lab	0	0	3	3	1.5
2	ENGG	Major	CS492	Computer Networks Lab	0	0	3	3	1.5
3	ENGG	Major	CS493	Machine Learning Lab	0	0	3	3	1.5
4	ENGG	Minor	M(CS)491	Introduction to R Programming	0	0	3	3	1.5
5	HUM	Ability Enhancement Course	HU(CS)491	Soft Skill & Aptitude	0	0	3	3	1.5
Total of Theory, Practical								**29**	**22.5**
Total Second Year Credit									**50.5**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Database Management Systems
Course Code: CS401
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites:

1. Logic of programming language
2. Basic concepts of data structure and algorithms

Course Objective(s):

- To learn the data models, conceptualize and depict a database system
- To design system using E-R diagram.
- To learn SQL & relational database design.
- To understand the internal storage structures using different file and indexing techniques.
- To know the concepts of transaction processing, concurrency control techniques and recovery procedure**.**

Course Outcome(s):

After completion of the course students will be able to

CO1: To express the knowledge of data models
CO2: To implement the concept of designing an efficient relational database system
CO3: To correlate real world queries with database system.
CO4: To illustrate transaction processing, concurrency control and recovery management of a database.
CO5: To assess the internal storage structure to implement a proper database for an organization.

Course Content:
Module I:
Introduction [3L]
Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS.

Module II:
Entity-Relationship and Relational Database Model [9L]
Basic concepts, Design Issues, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R features, case study on E-R Model. Structure of relational Databases, Relational Algebra, Relational Calculus, Extended Relational Algebra Operations, Views, Modifications of the Database.

Module III: [4L]
SQL and Integrity Constraints [6L]
Concept of DDL, DML, DCL. Basic Structure, set operations, Aggregate Functions, Null

Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Subqueries, Database security application development using SQL, Stored procedures and triggers

Module IV:
Relational Database Design [6L]
Functional Dependency, Different anomalies in designing a Database., Normalization using functional dependencies, Decomposition, Boyce-Codd Normal Form, 3NF, Normalization using multi-valued dependencies, 4NF, 5NF, Case Study

Module V:
Internals of RDBMS [6L]
Physical data structures, Query optimization: join algorithm, statistics and cost bas optimization. Transaction processing, Concurrency control and Recovery Management: transaction model properties, state serializability, lock base protocols; two phase locking, Dead Lock handling

Module VI:
File Organization & Index Structures [6L]
File & Record Concept, placing file records on Disk, Fixed and Variable sized Records, Types of Single-Level Index (primary, secondary, clustering), Multilevel Indexes

Text book:
1. Henry F. Korth and Silberschatz Abraham, “Database System Concepts”, McGraw Hill.

2.Elmasri Ramez and Navathe Shamkant, “Fundamentals of Database Systems”, Benjamin Cummings Publishing Company.

Reference book:
1. Jain: Advanced Database Management System Cyber Tech
2. Date C. J., “Introduction to Database Management”, Vol. I, II, III, Addison Wesley.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	2	-	-	3	-	-	3	2	2
CO2	3	3	3	2	3	2	-	-	2	-	-	3	2	2
CO3	3	3	3	3	3	-	-	-	2	-	-	3	1	1
CO4	2	2	2	2	3	1	-	-	2	-	-	3	1	2
CO5	3	2	2	2	3	1	-	-	1	-	-	3	2	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Computer Networks
Course Code: CS402
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites:
1. Familiarity and knowledge of Operating Systems and Computer Architecture.
2. Also require a little bit of programming languages concepts like C, Java.

Course Objective(s):
- To be familiar with the basics of data communication
- To be familiar with various types of computer networks
- To have experience in designing communication protocols
- To be exposed to the TCP/IP protocol suite

Course Outcome(s):
After completion of the course students will be able to

CO1: Understand basics of computer network and different architecture and topologies of computer network and analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
CO2: Understand/analyze different protocols of the data link layer and apply them to solve engineering problems.
CO3: Understand/analyze different protocols of Network and Transport Layer and apply them to solve engineering problems.
CO4: Understand/analyze different protocols of session and application layer and apply them to solve engineering problems.
CO5: Develop, Analyze, specify and design the topological and routing strategies using socket programming.

Course Content:
Module 1: Introduction and Physical Layer [7L]:
Introduction [3L]:
Computer Network, data communication, topology, OSI & TCP/IP Reference Models, layers and characteristics, Wireless Network, comparison to wired and wireless network.
Physical Layer: [4L]
Overview of data (analog & digital), signal (analog & digital), transmission (analog & digital) & transmission media (guided & unguided); Circuit switching: time division & space division switch, TDM bus; Telephone Network.

Module 2: Data Link Layer [10L]
Framing, Error Detection and Correction, Flow Control(stop and wait, Sliding window protocol, implementation of sliding window protocol using Go- Back-N and Selective

Repeat), HDLC, PPP. Medium Access Control Sub-layer, Multiple Access Protocols: ALOHA, Carrier Sense Multiple Access Protocols, IEEE 802.x Ethernet, Switched Ethernet, Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet, Wireless LANs - IEEE 802.xx, Bluetooth, RFID, Bridges, Virtual LANs, Switching.

Module 3: Network Layer [10L]
IP Addressing (IPv4 and IPv6, classful and classless addressing), Subletting, Super netting, Packet Switching (Virtual-Circuit and Datagram Networks), ARP, IP, ICMP, IPV6, BOOTP and DHCP–Delivery protocols Other Protocols such as mobile IP in wireless Network. [5L] Routing: Shortest Path Algorithms, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing, Any cast Routing: RIP, OSPF, BGP; Routing for Mobile Hosts. [5L]

Module 4: Transport layer: [5L]
Process to Process delivery; UDP; TCP, SCTP, TCP RENO, TCP/IP in Wireless environment, Congestion control in TCP: Congestion Control: Open Loop, Closed Loop packets; Quality of service: techniques to improve QoS: Leaky bucket algorithm, Token bucket algorithm.

Module 5: Application Layer [4L]
Introduction to DNS, SMTP, SNMP, FTP, HTTP & WWW: Cryptography (Public, Private Key based), Digital Signature, Firewalls

Text books:
1. B. A. Forouzan – Data Communications and Networking (3rd Ed.) —– TMH
2. S. Tanenbaum – —Computer Networks (4th Ed.)‖ – Pearson Education/PHI

Reference books:
1. W. Stallings – Data and Computer Communications (5th Ed.)‖ – PHI/ Pearson Education
2. Zheng & Akhtar, Network for Computer Scientists & Engineers, OUP
3. Comer – Internetworking with TCP/IP, vol. 1, 2, 3(4th Ed.)‖ – Pearson Education/PHI

CO–PO Mapping:

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

CO-PSO Mapping:

COs	PSO1	PSO2	PSO3
CO1	3	3	3
CO2	3	3	3
CO3	3	3	3
CO4	3	3	3
CO5	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Machine Learning
Course Code: CS403
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites:

1. Basic programming skills (preferably Python or Java)
2. Algorithm design and analysis
3. Probability: axioms, conditional probability, common distributions (Bernoulli, Binomial, Gaussian, etc.), expectation, variance, Bayes' rule
4. Linear algebra: vectors, matrices, eigenvalues/eigenvectors
5. Basic calculus and statistics
6. Data structures (arrays, lists, trees, graphs)

Course Objectives

1. To introduce the fundamental concepts of supervised and unsupervised learning, including real-world applications, model training workflows, data preprocessing, and evaluation metrics.

2. To provide practical understanding of clustering techniques, dimensionality reduction, and anomaly detection for unsupervised learning and pattern discovery.

3. To familiarize students with model evaluation strategies, cross-validation, and the theoretical concepts of bias-variance trade-off for robust model development.

4. To develop foundational knowledge in neural networks and deep learning architectures, including sequence modeling, transfer learning, and representation learning.

5. To explore scalable and modern machine learning paradigms such as online, semi-supervised, and reinforcement learning, along with probabilistic graphical models and distributed ML.

6. To expose students to recent trends in machine learning including Auto ML, federated learning, interpretability, and the ethical aspects of AI systems.

Course Outcomes (COs)

CO1. Understand the fundamental concepts and methods of machine learning
CO2. Learn to design and analyze various machine learning algorithms
CO3. Explore supervised and unsupervised learning paradigms
CO4. Gain practical experience with machine learning tools and libraries
CO5. Develop the ability to apply machine learning techniques to solve real-world problems

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content

Module 1: [8L]
Supervised learning basics, Types of machine learning, Real-world applications, ML workflow and pipeline, Data preprocessing and feature engineering, Train-test-validation split, Overfitting and regularization, Evaluation metrics

Module 2: [5L]
Clustering fundamentals, K-means algorithm, Hierarchical clustering, Dimensionality reduction (PCA), Anomaly detection

Module 3: [4L]
Model evaluation strategies, Cross-validation techniques, Confusion matrix and ROC, Bias-variance trade-off

Module 4: [7L]
Introduction to neural networks, Perceptron and activation functions, Feed-forward architectures, Deep learning overview, Feature representation learning, Sequence and time-series modeling, Transfer learning basics

Module 5: [7L]
Scalable machine learning, Online learning concepts, Distributed computing in ML, Semi-supervised learning, Active learning, Reinforcement learning introduction, Probabilistic graphical models

Module 6: [4L]
Recent trends in ML, Federated and privacy-preserving learning, AutoML and hyper parameter tuning, Interpretability and ethical AI

Textbook:

1. Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The Elements of Statistical Learning*, Springer
3. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (2 e), O'Reilly – added for its practical, beginner-friendly approach.

Reference

1. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007
2. Andriy Burkov, *The Hundred-Page Machine Learning Book*, 2019 – concise overview for quick revision.

CO–PO / PSO Mapping

CO\PO	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 10	PO1 11	PS O1	PS O2	PS O3
CS403.1	3	2	3	2	2	–	–	–	–	–	–	1	1	–
CS403.2	3	3	2	–	–	–	–	–	–	–	1	–	1	–
CS403.3	2	3	3	2	–	–	–	–	–	–	–	1	–	–
CS403.4	2	2	3	3	2	–	–	–	–	–	2	–	–	–
CS403.5	2	2	2	3	1	1	–	–	–	2	–	–	–	–

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Formal Language and Automata Theory
Course Code: CS404
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites:
1. Digital Logic
2. Computer organization
3. Computer Fundamentals

Course Objective(s):

1. To give an overview of the theoretical foundations of computer science from the perspective of formal languages
2. To illustrate finite state machines to solve problems in computing
3. To explain the hierarchy of problems arising in the computer sciences.
4. To familiarize Regular grammars, context frees grammar.

Course Outcomes:
On successful completion of the learning sessions of the course, the learner will be able:

CS404.1: To acquire the knowledge of the basics of state machines with or without output and its different classifications
CS404.2: To summarize synchronous sequential circuits as the foundation of digital system.
CS404.3: To implement different techniques of designing grammars and recognizers for several programming languages.
CS404.4: To measure Turing's Hypothesis as a foreword to algorithms.
CS404.5: To assess the power and limitation of a computer, and take decisions on computability.

Course Contents:

Module-1: [9L]
Fundamentals: Basic definition of sequential circuit, block diagram, mathematical representation, concept of transition table and transition diagram. **[1L]**
Introduction to Finite State Model (FSM), Design of sequence detector, Finite State Machine, Finite Automata, Deterministic Finite Automation (DFA) and Non-deterministic Finite Automation (NFA), Transition diagrams, Transition tables and Language recognizers. **[3L]**
NFA with empty transitions, Equivalence between NFA with and without empty transitions. NFA to DFA conversion. **[2L]**
Minimization of FSM: Minimization Algorithm for DFA, Introduction to Myhill-Nerode Theorem **[2L]**

Limitations of FSM, Application of Finite Automata. **[1L]**

Module-2: [7L]
Finite Automata with output − Moore & Mealy machine. Representation of Moore & Mealy Machine, Processing of the String through Moore & Mealy Machine, Equivalence of Moore & Mealy Machine –Inter-conversion.**[2L]**
Equivalent states and Distinguishable States, Equivalence and k-equivalence, Minimization of Mealy Machine **[1L]**
Minimization of incompletely specified machine − Merger Graph, Merger Table, Compatibility Graph [2L]
Lossless and Lossy Machine–Testing Table, Testing Graph **[2L]**

Module-3: [5L]
Regular Languages, Regular Sets, Regular Expressions, Algebraic Rules for Regular Expressions, Arden's Theorem statement and proof **[1L]**
Constructing Finite Automata (FA) for given regular expressions, Regular string accepted by FA **[2L]**
Constructing Regular Expression for a given Finite Automata **[1L]**
Pumping Lemma of Regular Sets. Closure properties of regular sets **[1L]**

Module-4: [9L]
Grammar Formalism - Context Free Grammars, Derivation trees, sentential forms. Rightmost and leftmost derivation of strings, Parse Tree, Ambiguity in context free grammars. **[1L]**
Minimization of Context Free Grammars. **[1L],** Removal of null and unit production **[1L]**
Chomsky normal form and Greibach normal form.**[1L]**
Pumping Lemma for Context Free Languages. **[1L]**
Closure property of CFL, Ogden's lemma & its applications **[1L],**
Regular grammars – right linear and left linear grammars **[1L]**
Pushdown Automata: Pushdown automata, definition. Introduction to DCFL, DPDA, NCFL, NPDA **[1L]**
Acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. **[1L]**
Equivalence of CFL and PDA, inter-conversion. **[1L]**

Module-5: [5L]
Turing Machine: Turing Machine, definition, model **[1L]**
Design of TM, Computable functions **[1L],** Church's hypothesis, counter machine **[1L]**
Types of Turing machines **[1L]**
Universal Turing Machine, Halting problem **[1L]**

Textbook:
1. "Introduction to Automata Theory Language and Computation", Hopcroft. E. and Ullman J.D., Pearson Education.

Reference Books:

1. "Formal Languages and Automata Theory", C.K. Nagpal, Oxford
2. "Switching and Finite Automata Theory", Zvi Kohavi, 2nd Edition, Tata McGrawHill

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CS404. 1	3	3	2	2	1	-	-	-	-	-	-	-	2	-
CS404. 2	2	3	2	3	2	-	-	-	-	-	1	-	2	-
CS404. 3	3	2	2	2	3	-	-	-	-	-	2	2	3	-
CS404. 4	2	2	3	2	3	-	-	-	-	-	1	2	2	-
CS404. 5	2	2	3	2	2	-	-	-	-	-	2	1	2	-

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: PROBABILITY AND STATISTICS
Course Code: M(CS)401
Contact: (L: T: P): 3: 0: 0
Total Contact Hours: 36
Credit: 3

Prerequisites:
The students to whom this course will be offered should have a fundamental understanding of (10+2) standard calculus, elementary probability, set theory, functions, and basic counting principles to effectively comprehend the probabilistic and statistical concepts.

Course Objective(s):
The objective of the course is to make the students able to –

1. Develop a strong foundation in probability theory and random variables to model computational uncertainty and data-driven phenomena.
2. Gain analytical skills in joint distributions, correlation, regression, and parameter estimation relevant to machine learning and data analysis.
3. Build the competence to apply statistical hypothesis testing in evaluating algorithms, systems performance, and experimental results.

Course Outcomes (COs):

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

CO1	Apply the concepts of probability distributions and random variables to model uncertainty in computational and data-driven systems.
CO2	Apply sampling theory and parameter estimation techniques to infer population characteristics and support data-driven decision-making.
CO3	Analyze bivariate data using joint, marginal, and conditional distributions, correlation, and linear regression to interpret relationships and dependencies in real-world datasets.
CO4	Analyze decision-making scenarios using hypothesis testing procedures to examine statistical assumptions and validate system performance.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CO1	3	2	-	-	-	-	-	-	-	-	1			
CO2	3	2	-	-	-	-	-	-	-	-	1			
CO3	3	3	1	1	-	-	-	-	-	-	2			
CO4	3	3	1	1	-	-	-	-	-	-	2			
M(CS)401	**3**	**2.5**	**1**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**1.5**			

Weightage Values: Strongly mapped: ‘3’, Moderately mapped: ‘2’, Weakly mapped: ‘1’, Not mapped: ’-‘.

Course Content:

Module-I: Probability and Random Variables **[10L]**

Discrete and continuous random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, Moments, Moment generating functions, Binomial, Poisson, Uniform, Exponential and Normal distributions

Module-II: Two Dimensional Random Variables **[9L]**

Joint distributions, Marginal and conditional distributions, Covariance, Correlation and linear regression, T Transformation of random variables, Central limit theorem (for independent and identically distributed random variables).

Module-III: Sampling Theory & Estimation of Parameters **[10L]**

Sampling Theory: Random sampling, Parameter & Statistic, Standard error of statistic, Distributions of the sample mean and the sample variance for a normal population, Chi-Square distributions, t distributions and F distributions.

Estimation of Parameters: Unbiased and consistent estimators, Point estimation, Interval estimation, Maximum likelihood estimation of parameters (Binomial, Poisson and Normal), Confidence intervals and related problems.

Module-IV: Testing of Hypothesis **[7L]**

Simple and Composite hypothesis, Critical region, Level of significance, Type I and Type II errors, one sample and two sample tests for means and proportions, $\chi 2$ - test for goodness of fit.

Text Books:

1. Probability and Statistics, Das, N.G., The McGraw Hill Companies.
2. Fundamentals of Mathematical Statistics, Gupta S. C. and Kapoor V. K., Sultan Chand & Sons.
3. Fundamental of Statistics, Goon A.M., Gupta M. K. and Dasgupta, B., The World Press Pvt. Ltd.
4. Advanced Engineering Mathematics, Kreyszig, E., 9thEdition; John Wiley & Sons, 2006.

Reference Books:

1. Schaum's Outline in Probability (2ndEd.), Lipschutz, S.and Lipson, M., McGraw Hill Education.
2. Fundamentals of Probability and Statistics for Engineers, Soong, T. T., Wiley Publications.
3. Theory and Problems of Probability and Statistics (Schaum's Outline Series), Spiegel, M. R., McGraw Hill Book Co.

4. Applied Statistics and Probability for Engineers, Montgomery, D.C. and Runger, G.C., Wiley Publications.

Paper name: Principles of Management
Paper Code: HU(CS)401
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Course Objective(s):

- Explain the fundamental functions of management—planning, organizing, leading, and controlling—and their interrelationships in achieving organizational goals.
- Analyze and apply classical, behavioral, and contemporary management theories to solve problems and make decisions in organizational contexts.
- Demonstrate the ability to make informed, ethical, and strategic decisions by integrating management principles with current business practices.
- Foster leadership, communication, and collaborative skills essential for effectively managing teams and driving performance in diverse work environments.

Course Outcome(s):
After completion of the course students will be able to

CO1: To recall and identify the relevance of management concepts.
CO2: To apply management techniques for meeting current and future management challenges faced by the organization
CO3: To compare the management theories and models critically to solve real life problems in an organization.
CO4: To apply principles of management in order to execute the role as a manager in an organization.

Course Content:
Module-1 (6L):
Management Concepts: Definition, roles, functions and importance of Management, Evolution of Management thought-contribution made by Taylor, Fayol, Gilbreth, Elton Mayo, McGregor, Maslow

Module - 2: Planning and Control (6L):
Planning: Nature and importance of planning, -types of planning, Levels of planning - The Planning Process. –MBO, SWOT analysis, McKinsey's7S Approach.
Organising for decision making: Nature of organizing, span of control, Organisational structure –line and staff authority.
Basic control process -control as a feedback system – Feed Forward Control –Requirements for effective control – control

Module - 3: Group dynamics(6**L)**:
Types of groups, characteristics, objectives of Group Dynamics.
Leadership: Definition, styles & functions of leadership, qualities for good leadership, Theories of leadership

Module – 4(6L): Work Study and work measurement: Definition of work study**,** Method Study Steps, Tools and Techniques used in the Method Study and Work Measurement Time Study: Aim & Objectives, Use of stopwatch procedure in making Time Study. Performance rating, allowances and its types. Calculation of Standard Time. Work sampling

Module - 5: Marketing Management (**4L)**:
Functions of Marketing, Product Planning and development, Promotional Strategy

Module - 6: Quality management (**8L)**: Quality definition, Statistical quality control, acceptance sampling, Control Charts –Mean chart, range chart,c chart,p chart,np chart, Zero Defects, Quality circles, Kaizen & Six Sigma, ISO -9000 Implementation steps, Total quality management

Text Books:

1. Essentials of Management, by Harold Koontz & Heinz Weihrich Tata McGraw
2. Production and Operations Management-K. Aswathapa, K. Shridhara Bhat, Himalayan Publishing House

References:

1. Organizational Behavior, by Stephen Robbins Pearson Education, New Delhi
2. New era Management, Daft, 11thEdition, Cengage Learning
3. Principles of Marketing, Kotlar Philip and Armstrong Gary, Pearson publication

CO-PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	-	-	2	-	-	-		3	-	3	-	1	1
CO2		-	-	-	-	2	-	3	3	-	3	-	1	1
CO3	2	-	-	2	-	-	-		2	-	3	-	1	1
CO4	3	-	-	-	-	2	-	2	3	-	3	-	1	1

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Database Management Systems Lab
Course Code: CS491
Contact (Periods/Week): 3:0:0
Credit Point: 1. 5
No. of Lectures: 36

Prerequisites:

1. Logic of programming language
2. Basic concepts of data structure and algorithms

Course Objective(s):

- To learn the data models, conceptualize and depict a database system
- To learn the fundamental concepts of SQL queries.
- To understand the concept of designing a database with the necessary attributes.
- To know the methodology of Accessing, Modifying and Updating data & information from the relational databases.
- To learn database design as well as to design user interface and how to connect with the database.

Course Outcome(s):

After completion of the course students will be able to

CO1: To understand the basic concepts regarding database, know about query processing and techniques involved in query optimization and understand the concepts of database transaction
and related database facilities including concurrency control, backup and recovery.
CO2: To understand the introductory concepts of some advanced topics in data management like distributed databases, data warehousing, deductive databases and be aware of some advanced databases like partial multimedia and mobile databases.
CO3: To differentiate between DBMS and advanced DBMS and use of advanced database concepts and become proficient in creating database queries.
CO4: To analyse database system concepts and apply normalization to the database.
CO5: To apply and create different transaction processing and concurrency control applications.

Course Content:
Module I: [6L]
Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specifying constraints etc.).

Module II: [3L]
Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, represent attributes as columns, identifying keys) and apply the normalization techniques.

Module III: [6L]
Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables

Module IV: [3L]
Practicing DML commands- Insert, Select, Update, Delete

Module V: [6L]
Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc., Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Module VI: [6L]
Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping, Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger

Module VII: [6L]
Procedures- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure, PL/SQL, Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

Text book:
1. SQL, PL/SQL by Ivan Bayross, BPB Publications

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	2	-	-	3	-	-	3	2	2
CO2	3	3	3	2	3	2	-	-	2	-	-	3	2	2
CO3	3	3	3	3	3	-	-	-	2	-	-	3	1	1
CO4	2	2	2	2	3	1	-	-	2	-	-	3	1	2
CO5	3	2	2	2	3	1	-	-	1	-	-	3	2	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Computer Networks Lab
Course Code: CS492
Contact (Periods/Week): 0:0:3
Credit Point: 1.5
No. of Lectures: 36

Prerequisites:

1. Familiarity and knowledge of Computer Network and Computer Architecture
2. Also require strong knowledge of programming languages like C, Java and UNIX or Linux environment**.**

Course Objectives

1. To introduce students to the UNIX/Linux environment with a focus on networking commands, internetworking concepts, and hardware setup including cable crimping and configuration.

2. To develop the ability to implement and test network communication using socket programming over TCP and UDP protocols.

3. To provide hands-on experience in implementing routing protocols such as RIP and OSPF to understand dynamic routing mechanisms.

4. To familiarize students with network simulation tools like Packet Tracer, NS2/NS3, OMNeT++, and TinyOS for modeling and analyzing network behaviors.

5. To equip students with the skills to configure a basic web server and understand server-side networking principles.

Course Outcome(s):

After completion of the course students will be able to

CO1: To design and implement small size network and to understand various networking commands.

CO2: To provide the knowledge of various networking tools and their related concepts.

CO3: To understand various application layer protocols for its implementation inclient/server environment.

CO4: Understand the TCP/IP configuration for Windows and Linux.

CO5: Learn the major software and hardware technologies used on computer networks.

Course Content:

1. Familiarization of UNIX or Linux environment, UNIX or Linux general Commands specially Network Commands. Familiarization of Internetworking - Network Cables - Color coding - Crimping. Internetworking Operating Systems - Configurations. [6L]
2. Socket Programming using TCP and UDP [18L]
3. Implementing routing protocols such as RIP, OSPF. [3L]
4. Familiarization of advanced simulators like Packet Tracer, NS2/NS3, OMNET++, TinyOS [3L]
5. Server Configuration: only web server (If time permit, Instructor can do more than that) [6L]

Textbooks:

1. TCP socketsin C Programs-Practical guide for Programmers by Micheal,J Donahoo and Kenneth L calvert.
2. Socket Programming by Raj Kumar Buyaa.

CO–PO-PSO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3								2	2	3
CO2	3	3	3	3								2	3	3
CO3	3	3	3	3								3	3	3
CO4	3	3	3	3								3	2	3
CO5	3	3	3	3								3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Machine Learning Lab
Course Code: CS493
Contact (Periods/Week): 0:0:3
Credit Point: 1.5
No. of Lectures: 36

Course Objective(s):

This course will enable students to

- Make use of data sets in implementing machine-learning algorithms.
- Implement machine-learning concepts and algorithms in any suitable programming language.

Course Outcome(s):
After completion of the course students will be able to

CO1: Understand the implementation procedures for machine-learning algorithms.
CO2: Design Java/Python programs for various learning algorithms.
CO3: Apply appropriate data sets to machine-learning algorithms
CO4: Identify and apply machine-learning algorithms to solve real-world problems.

List of Lab Experiments

1. Implement and demonstrate the **FIND-S** algorithm for finding the most specific hypothesis based on a given set of training data samples (read data from a CSV file).
2. For a given set of training examples stored in a CSV file, implement and demonstrate the **Candidate-Elimination** algorithm to output all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision-tree-based **ID3** algorithm. Use an appropriate data set to build the tree and classify a new sample.
4. Build an **Artificial Neural Network** by implementing the back-propagation algorithm and test it with an appropriate data set.
5. Write a program to implement the **naïve-Bayes classifier** for a sample training data set stored as a CSV file. Compute classifier accuracy on a test set.
6. Assuming a set of documents that need classification, use a naïve-Bayes model to perform this task (built-in Java/Python libraries may be used). Calculate accuracy, precision and recall.
7. Construct a **Bayesian network** for medical data and demonstrate diagnosis of heart patients using the standard Heart-Disease data set (use Java/Python ML libraries).
8. Apply the **EM algorithm** to cluster data stored in a CSV file. Use the same data set for clustering with **k-means**; compare results and comment on clustering quality (libraries may be used).

9. Implement the **k-Nearest-Neighbour** algorithm to classify the Iris data set. Print both correct and wrong predictions (libraries permitted).
10. Implement **Locally Weighted Regression** (non-parametric) to fit data points. Select an appropriate data set and draw graphs of the fit.
11. **Build and evaluate a logistic-regression classifier** on a binary data set (e.g., Breast-Cancer or Spam). Report accuracy and ROC curve.
12. **Perform principal-component analysis (PCA)** for dimensionality reduction on a high-dimensional data set, then apply k-means to the reduced data and visualise the clusters.

CO–PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CS493.1	3	4	4	–	–	–	–	–	–	–	3
CS493.2	–	4	4	3	4	3	–	–	–	–	–
CS493.3	–	3	–	3	3	3	–	–	–	–	–
CS493.4	3	4	–	4	–	3	–	–	–	–	–

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

COURSE NAME: INTRODUCTION TO R PROGRAMMING
COURSE CODE: M(CS)491
CONTACT (PERIODS/WEEK): 0:0:3
CREDIT POINT: 1.5

PREREQUISITES:

- Basic knowledge of programming concepts (preferably in Python, C, or any high-level language)
- Fundamentals of statistics and mathematics
- Familiarity with basic data structures and file handling concepts

COURSE OBJECTIVE(S): To introduce students to the fundamentals of R programming for data handling, statistical analysis, and effective data visualization.

COURSE OUTCOME(S): After completion of the course students will be able to

CO1: To understand and apply the fundamental concepts, syntax, and data structures of R programming for statistical computing.

CO2: To import, clean, transform, and manage diverse datasets using built-in R functions and libraries such as dplyr and tidyr.

CO3: To perform exploratory data analysis and generate meaningful statistical summaries for data-driven insights.

CO4: To create effective visualizations using base R graphics and ggplot2 to communicate analytical findings.

CO5: To implement basic statistical models and machine learning algorithms in R for data interpretation and predictive analysis.

COURSE CONTENT:

Week 1: Introduction to R and RStudio

- Installing R and RStudio
- R Console, Scripts, Packages
- Basic Syntax: Variables, Data Types, Operators
- Introduction to Vectors

  Lab: Simple R programs, using RStudio interface

Week 2: Data Structures in R

- Vectors, Lists, Matrices, Arrays
- Factors and Data Frames
- Indexing and Subsetting

  Lab: Creating and manipulating various data structures

Week 3: Control Structures and Functions

- Conditional Statements: if, else, switch
- Loops: for, while, repeat

- Writing Custom Functions
- Scope of Variables

Lab: Writing loops and functions for data manipulation

Week 4: Data Import and Export

- Reading Data: CSV, Excel, Text, JSON
- Writing Data to Files
- Web Data Reading
- Using readr, readxl, jsonlite

Lab: Loading and saving datasets using various formats

Week 5: Data Cleaning and Manipulation

- Handling Missing Data
- Renaming Columns, Filtering, Sorting
- dplyr package: filter(), select(), mutate(), summarise()

Lab: Cleaning and manipulating real-world datasets

Week 6: Exploratory Data Analysis (EDA)

- Descriptive Statistics
- Grouped Summaries
- Data Transformation and Aggregation

Lab: EDA using summary(), aggregate(), tapply()

Week 7: Data Visualization with Base R

- Base Graphics: plot(), hist(), boxplot(), barplot()
- Customizing Plots: Titles, Labels, Colors
- Saving Plots to File

Lab: Plotting datasets using base graphics

Week 8: Data Visualization with ggplot2

- Introduction to ggplot2 Grammar
- Creating Aesthetic Mappings
- Plot Types: Line, Bar, Scatter, Boxplot, Heatmap
- Themes and Faceting

Lab: Creating multi-layered plots using ggplot2

Week 9: Working with Dates and Strings

- Date and Time Handling with lubridate
- String Operations with stringr

Lab: Text and date processing examples

Week 10: Statistical Computing in R

- Probability Distributions: Normal, Binomial, Poisson
- Hypothesis Testing: t-test, chi-square, ANOVA
- Correlation and Regression Analysis

Lab: Conducting tests using stats package

Week 11: Introduction to Machine Learning with R

- Overview of ML in R
- Simple Linear Regression
- Logistic Regression
- Using caret and mlr packages

Lab: Build and evaluate a basic predictive model

Week 12: Capstone Mini Project & Revision

- Students choose a dataset
- Apply data cleaning, visualization, and modeling
- Report and Presentation
  Lab: Project implementation and submission

CO–PO MAPPING:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	-	-	-	-	-	-	-	3	3	3
CO2	2	3	3	3	-	-	-	-	-	-	-	3	3	3
CO3	3	3	2	2	-	-	-	-	-	-	-	3	3	3
CO4	3	2	3	3	-	-	-	-	-	-	-	3	3	3
CO5	3	3	2	3	-	-	-	-	-	-	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Soft Skills and Aptitude
Course Code: HU(CS)491
Contact: 2:0:0
Total Lectures: 24
Credits: 1

Course Outcomes:
On completion of the course students will be able:
CO1: Develop effective communication skills (verbal, written, and non-verbal) for professional environments.
CO2: Enhance problem-solving and analytical thinking through logical reasoning and aptitude training.
CO3: Demonstrate strong teamwork, leadership, and interpersonal skills in a corporate setting.
CO4: Manage time efficiently and handle workplace stress through practical strategies.
CO5: Build a professional profile (resume, LinkedIn, portfolio) to enhance employability.

Course Contents:

Module 1: Communication & Corporate Etiquette (5L)

Objective: To develop professional communication skills and workplace etiquette.

Topics Covered:
- Fundamentals of Corporate Communication (Email, Meetings, Reports)
- Business Writing: Resume, Cover Letter, and Statement of Purpose (SOP)
- Public Speaking & Presentation Skills (With AI/Tech-based Tools)
- Group Discussion & Personal Interview Techniques
- Corporate Etiquette: Dress Code, Body Language, Networking
- Persuasion & Negotiation Skills in Interviews
- Handling Workplace Conflicts and Professional Conduct

Practical Exercises:
- Mock GD & PI Sessions
- Elevator Pitch Challenge
- Drafting Business Emails and Reports
- Role-playing Interview Scenarios

Module 2: Verbal & Analytical Ability for Competitive Exams (4L)

Objective: To strengthen verbal reasoning, numerical ability, and logical thinking for aptitude tests.

Topics Covered:
- Grammar & Sentence Correction
- Vocabulary Building & Contextual Usage
- Reading Comprehension Techniques

- Logical Reasoning: Syllogisms, Blood Relations, Seating Arrangements
- Quantitative Aptitude: Percentage, Ratio-Proportion, Time & Work
- Data Interpretation & Graphical Analysis
- Puzzles and Critical Thinking Exercises
- Decision-Making and Problem-Solving Strategies

Practical Exercises:
- Weekly Verbal & Quantitative Aptitude Tests Logical Puzzles and Reasoning Games
- Speed Reading & Summary Writing Mock Corporate Aptitude Tests

Module 3: Teamwork, Leadership & Conflict Resolution (5L)

Objective: To cultivate leadership, teamwork, and interpersonal effectiveness.

Topics Covered:
- Teamwork vs. Individual Contribution
- Leadership Styles & Decision Making
- Conflict Management & Negotiation Skills
- Emotional Intelligence (EQ) in Workplace Relationships
- Cross-Cultural Communication in Global Tech Companies
- Adapting to Change & Handling Difficult Conversations
- Empathy & Relationship Management in Professional Spaces

Practical Exercises
- Team-Based Problem-Solving Activities
- Leadership Role-Play Scenarios
- Conflict Resolution Case Studies
- Workplace Adaptability Drills

Module 4: Time & Stress Management for Engineers (3L)

Objective: To develop productivity habits for balancing work and personal growth.

Topics Covered:
- Prioritization Techniques (Eisenhower Matrix, Pomodoro Technique)
- Handling Work Pressure & Burnout in IT Jobs
- Effective Goal-Setting & Productivity Hacks
- Mindfulness & Stress Reduction Techniques
- Work-Life Balance for Engineers
- Managing Deadlines & Overcoming Procrastination
- Stress-Handling Strategies in High-Stakes Environments

Practical Exercises:
- Time-Blocking & Task Prioritization Drills
- Stress-Management Workshops

- Industry Case Studies on Work-Life Balance
- Productivity Habit Challenges

Module 5: Profile Building & Career Readiness (7L)

Objective: To prepare students for industry placements and professional networking.

Topics Covered:
- Resume Optimization (ATS-Friendly Formats)
- LinkedIn & GitHub Profile Enhancement
- Personal Branding & Online Portfolio Development
- Industry Certifications & Upskilling (Coursera, Udemy, AWS, etc.)
- Networking Strategies & Building Corporate Connections
- Strategies for Effective Salary Negotiation
- Leveraging Internship & Freelance Experience for Career Growth

Practical Exercises:
- Resume Review & LinkedIn Profile Audit
- Creating a Personal Website/Portfolio
- Mock HR & Technical Interviews
- Industry Networking & Mentorship Sessions

Evaluation & Assessment
- Class Participation & Practical Assignments – 20%
- Aptitude & Verbal Ability Tests – 20%
- Mock GD & PI Performance – 20%
- Corporate Project & Leadership Activity – 20%
- Final Profile Submission (Resume, LinkedIn, Portfolio) – 20%

Industry Integration & Certifications
- Guest Lectures from Corporate Trainers, HR Leaders & Entrepreneurs
- Collaboration with Placement Cells & Corporate Partners (e.g., TCS, Infosys, Wipro)
- Soft Skill Certifications (TCS iON, NPTEL, LinkedIn Learning)

CO-PO mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
HU(CS)491.1	3	-	2	-	-	-	-	-	-	-	3	3	-	3
HU(CS)491.2	3	-	2	-	-	-	-	-	-	-	-	3	3	-
HU(CS)491.3	-	-	3	-	-	-	-	-	-	-	3	2	-	3
HU(CS)491.4	-	-	-	-	1	-	-	-	-	-	-	-	3	2
HU(CS)491.5	-	-	3	-	3	-	-	-	-	-	3	1	-	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

3rdYear5thSemester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS501	Software Engineering	3	0	0	3	3
2	ENGG	Major	CS502	Object Oriented Programming using Java	3	0	0	3	3
3	ENGG	Major	CS503A	Compiler Design	3	0	0	3	3
			CS503B	Cryptography and Network Security					
			CS503C	Computer Graphics					
			CS503D	Data Handling and Visualization					
4	ENGG	Major	CS504	Soft Computing	3	0	0	3	3
5	HUM	Minor	HU(CS)501	Project Management & Finance	2	0	0	2	2
B. PRACTICAL									
1	ENGG	Major	CS591	Software Engineering Lab	0	0	3	3	1.5
2	ENGG	Major	CS592	Object Oriented Programming using Java Lab	0	0	3	3	1.5
3	ENGG	Major	CS593A	Compiler Design Lab	3	0	0	3	1.5
			CS593B	Cryptography and Network Security Lab					
			CS593C	Computer Graphics Lab					
			CS593D	Data Handling and Visualization Lab					
4	ENGG	Major	CS594	Soft Computing Lab	0	0	3	3	1.5
5	PRJ	Project	CS582	Project-I	0	0	4	4	2
Total of Theory, Practical								**30**	**22**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Software Engineering
Course Code: CS 501
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites: Programming for Problem Solving

Course Objectives

- **Understand Software Development Life Cycles (SDLCs):** To equip students with knowledge of different SDLC models (e.g., Waterfall, Agile, Spiral) and their appropriate applications in software project planning and management.
- **Master Requirements Engineering:** To develop skills in gathering, analyzing, documenting, and validating software requirements using structured and systematic approaches.
- **Learn Software Design Principles:** To introduce students to architectural and design principles, including modularity, abstraction, design patterns, and modeling with UML.
- **Apply Software Testing and Quality Assurance Techniques:** To train students in verification, validation, and various software testing methodologies (unit, integration, system, acceptance testing) to ensure software reliability and quality.
- **Understand Project Management and Maintenance:** To provide knowledge of software project management practices, including estimation, scheduling, risk management, and maintenance activities post-deployment.

Course Outcomes:

On completion of the course students will be able

CS501.1: To identify software engineering problems, including specification, design, implementation, and testing of software systems that meet performance and quality assurance

CS501.2: To gather software requirements through a productive working relationship with various stakeholders of the project

CS501.3: To prepare solutions in one or more application domains using software engineering approaches that integrates ethical, social, legal and economic concerns.

CS501.4: To validate the code from the design and effectively apply relevant standards and perform testing, and quality management and practice.

CS501.5: To build modern engineering tools necessary for project management, software reuse and maintenance.

Course Content:

Module-1: [6L]

Introduction: Software Engineering, Characteristics, Components, Application, Definitions. Software Project Planning-Feasibility Analysis, Technical Feasibility, Cost-Benefit Analysis, Basics of estimation: COCOMO (Basic, intermediate, Complete) model.

Module- 2: [6L]

Evolution and impact of Software engineering, software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification.

Module -3: [8L]

Basic issues in software design, modularity, cohesion, coupling and layering, function-oriented software design: DFD and Structure chart, object modeling using UML, Object-oriented software development, user interface design. Coding standards and Code review techniques.

Module -4: [7L]

Fundamentals of testing, White-box, and black-box testing, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics, reliability growth modeling.

Module -5: [9L]

Software project management, Project planning and control, cost estimation, project scheduling using PERT and GANTT charts, cost-time relations: Rayleigh-Norden results, quality management, ISO and SEI CMMI, PSP and Six Sigma. Computer aided software engineering, software maintenance, software reuse, Component-based software development.

Text Books:

1. Fundamentals of Software Engineering by Rajib Mall, –PHI-3rd Edition, 2009.
2.Software Engineering-Pankaj Jalote(Wiley-India)

Reference Books:

1.Software Engineering–Agarwal and Agarwal(PHI)
2.Software Engineering, by Ian Sommerville, Pearson Education Inc., New Delhi, (2009).
3. Software Engineering: A Practitioner's Approach", by Roger S. Pressman, McGraw-Hill.(2005)

CO–PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	3	3	3	3	2	1	1	-	3	2	2	3	3	3
CO2	3	3	2	3	3	1	1	-	3	3	2	2	3	2
CO3	2	3	2	3	3	-	2	3	3	2	2	3	3	3
CO4	2	3	2	3	2	-	1	-	3	2	2	3	3	3
CO5	3	2	3	3	3	1	1	-	3	1	3	2	3	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Object Oriented Programming using Java
Course Code: CS502
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36
Prerequisite: Partial Object-Oriented Programming using C++

Course Objective(s):

- To introduce the foundational principles of Object-Oriented Programming and its implementation using Java, emphasizing real-world modeling and software design.
- To develop proficiency in Java programming through hands-on experience with core concepts like classes, inheritance, polymorphism, exception handling, and multithreading.
- To equip students with the ability to construct modular, reusable, and efficient Java applications, including GUI and applet-based programs.

Course Outcomes:
After completion of the course students will be able to
CO1: To define the process of interaction between Objects and System w.r.t. Object Oriented Paradigm.
CO2: To summarize basic concepts of Object Orientation in Java Programming along with different properties and features.
CO3: To implement various string handling functions as well as basic I/O operations in object-oriented environment.
CO4: To explain basic code reusability concept w.r.t. Inheritance, Package and Interface.
CO5: To construct Java programs utilizing core object-oriented concepts, with a focus on Exception Handling, Multithreading, and Applet-based Web Programming.

Course Contents:

Module I: Introduction [2L]
Object Oriented Analysis (OOA) & Object-Oriented Design (OOD) - Concepts of object-oriented programming language, Relationships among objects and classes-Generalization, Specialization, Aggregation, Association, Composition, links, Meta-class, Object Oriented Programming concepts - Difference between Java and C++; Different features of Java.

Module II: Java Basics [10L]
Basic concepts of java programming - Advantages of java, Byte-code & JVM, Data types, Different types of Variables, Java Operators & Control statements, Java loops, Array, Creation of class, object, method, Constructor- Definition, Usage of Constructor, Different types of Constructors, finalize method and garbage collection, Method & Constructor overloading, this keyword, use of objects as parameter & methods returning objects, Call by value & call by reference, Static variables & methods, Nested & inner classes.

Module III: Basic String handling & I/O [5L]
Basic string handling concepts- Concept of mutable and immutable string, Methods of String class, Methods of String buffer class, Command line arguments, basics of I/O operations – keyboard input using Buffered Reader, Scanner class in Java I/O operation.

Module IV: Inheritance and Java Packages [8L]
Inheritance - Definition, Advantages, Different types of inheritance and their implementation, Super and final keywords, super() method, Method overriding, Dynamic method dispatch, Abstract classes & methods, Interface - Definition, Use of Interface, Multiple inheritance by using Interface, Java Packages - Definition, Creation of packages, Java Access Modifiers - public, private, default and protected, Importing packages, member access for packages.

Module V: Exception handling, Multithreading and Applet Programming [11L]
Exception handling - Basics, different types of exception classes. Difference between Checked & Unchecked Exception, Try & catch related case studies, Throw, throws & finally, Creation of user defined exception, Multithreading - Basics, main thread, Thread life cycle, Creation of multiple threads - yield(), suspend(), sleep(n), resume(), wait(), notify(), join(), is Alive(),Thread priorities, thread synchronization, Interthread communication, deadlocks for threads, Applet Programming - Basics, applet life cycle, difference between application & applet programming, Parameter passing in applets.

Text book:
1. Herbert Schildt – "Java: The Complete Reference" – 9th Ed. – TMH
2. E. Balagurusamy – "Programming with Java: A Primer" – 3rd Ed. – TMH.

Reference
1. R.K. Das – "Core Java for Beginners" – VIKAS PUBLISHING.
2. Rambaugh, James Michael, Blaha – "Object Oriented Modelling and Design" – Prentice Hall, India.

CO-PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	3	-	-	-	-	-	3	2	2	3
CO2	2	3	3	2	2	-	-	-	-	-	2	2	2	3
CO3	3	3	2	2	1	-	-	-	-	-	1	2	2	3
CO4	2	2	3	2	3	-	-	-	-	-	2	2	2	3
CO5	3	3	2	3	1	-	-	-	-	-	1	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Compiler Design
Course Code: CS503A
Contact: 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:
1. Mathematics
2. Concept of programming languages
3. Data structures
4. Computer architecture
5. Formal languages and automata theory
6. Some advanced math might be required if you adventure in code optimization

Course Objective(s):
- To make the student understand the process involved in a compiler
- To create an overall view of various types of translators, linkers, loaders, and phases of a compiler
- To understand the concepts of syntax analysis, various types of parsers especially the top down approach
- To create awareness among students about various types of bottom-up parsers
- To understand the syntax analysis and, intermediate code generation, type checking, the role of symbol table and its organization, Code generation, machine in dependent code optimization and instruction scheduling

Course Outcomes:
On completion of the course students will be able
CO1: To define compilers and various components of a compiler.
CO2: To understand the roles of different phases of a compiler.
CO3: To implement different algorithms for designing different phases of compilers.
CO4: To build simple compilers using C programming languages.

Course Contents:
Module-1: Introduction to Compiler and Lexical Analyzer [7L]
Compilers, Cousins of the Compiler, Analysis-synthesis model, Phases of the compiler, Role of the lexical analyzer, Tokens, Patterns, Lexemes, Input buffering, Specifications of a token, Recognition of tokens, Finite automata, From a regular expression to an NFA, From a regular expression to DFA, Design of a lexical analyser generator (Lex).

Module-2: Syntax Analyzer and Semantic Analyzer [10L]
The role of a parser, Context free grammars, Writing a grammar, Top down Parsing, No recursive Predictive parsing (LL),Bottom up parsing, Handles, Viable prefixes, Operator precedence parsing, LR parsers (SLR, LALR, Canonical LR), Parser generators (YACC), Error Recovery strategies for different parsing techniques, Syntax directed translation: Syntax

directed definitions, Construction of syntax trees, Bottom-up evaluation of S-attributed definitions, L-attributed definitions, Bottom-up evaluation of inherited attributes.

Module-3: Type Checker and Run-time Environment [7L]
Type systems, Specification of a simple type checker, Equivalence of type expressions, Type conversions, Source language issues (Activation trees, Control stack, scope of declaration, Binding of names), Symbol tables, dynamic storage allocation techniques.

Module-4: Intermediate Language Generator [4L]
Intermediate languages, Graphical representation, Three-address code, Implementation of three address statements (Quadruples, Triples, Indirect triples).

Module-5: Code Optimizer and Code Generator [8L]
Consideration for Optimization, scope of optimization, local optimization, loop optimization, folding, DAG representation, Flow Graph, Data flow equation, global optimization, redundant sub expression elimination, induction variable elimination, copy propagation, basic blocks & flow graphs, transformation of basic blocks, DAG representation of basic blocks, peep hole optimization, Object code forms, machine dependent code optimization, register allocation and assignment, generic code generation algorithms, DAG for register allocation.

Text Books:
1. Aho, A. V., Sethi, R., & Ullman, J. D. Addison - Wesley, 2007. Compilers-Principles, Techniques, and Tools.
2. Holub, A. I. (1990). Compiler design in C (Vol.5). Englewood Cliffs, N J: Prentice Hall.

Reference Books:
1. Chattopadhyay, S. (2005). Compiler Design. PHI Learning Pvt. Ltd.
2. Tremblay, J. P., & Sorenson, P.G. (1985). Theory and Practice of Compiler Writing. McGraw-Hill, Inc.
3. Appel, A.W. (2004). Modern compiler implementation in C. Cambridge university press.
4. Barrett, W.A., Bates, R.M., Gustafson, D.A., & Couch, J.D. (1986). Compiler construction: theory and practice. SRA School Group.

CO-PO-PSO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	-	2	3	3	-	-	1	-	-	-	-	-	2	2
CO2	3	-	1	2	-	2	-	-	-	1	-	1	-	3
CO3	2	3	-	3	-	-	2	-	1	-	-	1	2	-
CO4	2	3	3	2	1	-	-	2	-	-	1	-	2	2
CO5	2	3	2	-	-	1	-	1	-	2	-	2	-	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Cryptography and Network Security
Course Code: CS503B
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites

1. Knowledge of Computer Networks and Operating Systems fundamentals
2. Understanding of Discreet Mathematics concepts

Course Objective(s):

- To impart concepts on cryptography and Network security
- To gain knowledge of the standard algorithms used to provide confidentiality, integrity, and authenticity
- To recognize the various key distribution and management systems for security of a cryptosystem

Course Outcomes:

After completion of course, students would be able

CS503B.1: To understand fundamental network security concepts, including services, classical encryption, and number theory, to analyze basic cryptographic mechanisms.

CS503B.2: To apply symmetric and asymmetric encryption algorithms, such as DES, AES, RSA, and elliptic curve cryptography, to design secure communication systems and manage cryptographic keys.

CS503B.3: To analyze and evaluate authentication mechanisms, hash functions, and digital signature schemes to ensure integrity, authenticity, and non-repudiation in data communication.

CS503B.4: To design and assess secure network architectures using firewalls, intrusion detection systems, and trusted systems, and implement practical countermeasures against various threats and attacks.

CS503B.5: To integrate and apply security protocols for email, IP, and web applications (e.g., PGP, S/MIME, IPSec, SSL/TLS) to achieve end-to-end secure communications in real-world scenarios.

Course Contents

Module-1: Introduction [7L]

Introduction - Services, Mechanisms, and Attacks, OSI security architecture, Network security model

Classical Encryption techniques (Symmetric cipher model, substitution techniques, transposition techniques, steganography)

Finite Fields and Number Theory: Groups, Rings, Fields, Modular arithmetic, Euclid's algorithm

Polynomial Arithmetic, Prime numbers, Fermat's and Euler's theorem

Testing for primality -The Chinese remainder theorem - Discrete logarithms

Module-2: Symmetric and Asymmetric Encryption Techniques [9L]
Data Encryption Standard- Block cipher principles, block cipher modes of operation
Advanced Encryption Standard (AES), Triple DES, Blowfish, RC5 algorithm
Public key cryptography: Principles of public key cryptosystems, The RSA algorithm
Key management - Diffie Hellman Key exchange, Elliptic curve arithmetic, Elliptic curve cryptography

Module-3: Authentication, Hashing, and Digital Signatures [6L]
Authentication requirement, Authentication function, MAC, Hash function
Security of hash function and MAC, MD5, SHA, HMAC, CMAC
Digital signature and authentication protocols, DSS, ElGamal, Schnorr

Module-4: Applied Network Security and Trusted Systems [7L]
Authentication applications, Kerberos, X.509
Internet Firewalls for Trusted System: Roles of Firewalls, Firewall related terminology- Types of Firewalls, Firewall designs principles
SET for E-Commerce Transactions
Intruder, Intrusion detection system
Virus and related threats, Countermeasures
Trusted systems, Practical implementation of cryptography and security

Module-5: Email, IP, and Web Security Protocols [7L]
E-mail Security: Security Services for E-mail-attacks possible through E-mail, Establishing keys privacy, authentication of the source
Message Integrity, Non-repudiation, Pretty Good Privacy, S/MIME
IP Security: Overview of IPSec, IPv4 and IPv6-Authentication Header, Encapsulation Security Payload (ESP)
Internet Key Exchange (Phases of IKE, ISAKMP/IKE Encoding)
Web Security: SSL/TLS Basic Protocol, computing the keys, client authentication
PKI as deployed by SSL Attacks fixed in v3, Exportability, Encoding, Secure Electronic Transaction

Text Books:
1. Kahate, A. (2013). Cryptography and network security. Tata McGraw-Hill Education.
2. Forouzan, B. A., & Mukhopadhyay, D. (2015). Cryptography and network security. New York, NY: Mc Graw Hill Education (India) Private Limited.

Reference Books:
1. Stallings, W. (2006). Cryptography and network security, 4/E. Pearson Education India.
2. Daras, N. J., &Rassias, M. T. (Eds.). (2015). Computation, cryptography, and network security (pp. 253-287).Springer.
3. Kumar, A., & Bose, S. (2017). Cryptography and network security. Pearson Education India.

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS503B.1	3	3	2	1	1	-	-	-	3	2	1	3	1	1
CS503B.2	3	2	3	2	3	-	-	-	2	1	1	3	1	3
CS503B.3	1	3	2	3	2	-	-	-	2	3	1	3	1	1
CS503B.4	2	3	1	3	1	-	-	-	1	1	1	3	1	1
CS503B.5	2	3	3	3	3	-	-	-	1	1	1	3	1	1

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Computer Graphics
Course Code: CS**503C**
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites:
Mathematics, Computer Language like C, C++ and Basic Principle of Computer Programming.

Course Objective(s)

- Understand basic principles and technologies of computer graphics.
- Implement geometric and algorithmic graphics techniques.
- Apply clipping and projection for object visualization.
- Generate and manipulate curves in graphics.
- Use hidden surface removal methods for 3D rendering.

Course Outcome(s)
On successful completion of the learning sessions of the course, the learner will be able
CO1: To remember the foundations of computer graphics and different display technology and devices.
CO2: To analyze the concept of geometric, mathematical and algorithmic approach necessary for programming computer graphics.
CO3: To explain clipping with the comprehension of windows, view-ports in relation to images display on screen.
CO4: To experiment and compare different hidden surface illumination methods

Course Content:
Module-I: Introduction to computer graphics [5L]:
Overview of computer graphics, Basic Terminologies in Graphics, lookup table,3D viewing devices, Plotters, printers, digitizers, light pens etc., Active & Passive graphics, Computer graphics software. Light &Color models, Raster Scan and Random scan displays.

Module - II: Scan conversion: [10L]
Points & lines, Line drawing algorithms; DDA algorithm, Bresenham's line algorithm, Midpoint Circle generation algorithm, Ellipse generating algorithm. Scan line polygon fill algorithm, boundary fill algorithm, flood fill algorithm

Module - III:2D and 3D Transformation [8L]
Basic transformations: translation, rotation, scaling, Matrix representations & homogeneous coordinates, transformations between coordinate systems, reflection shear, Transformation of points, lines, parallel lines, intersecting lines.
3D transformations: translation, rotation, scaling.

Module – IV: 2D-Viewing, Clipping and Projection [8L]
Viewing pipeline, Window to viewport co-ordinate transformation. Clipping operations: Point clipping, Cohen Sutherland line clipping algorithm, Polygon clipping algorithm, Viewport clipping, Basic concepts of different type of projections.

Module – V: Curve generation and Hidden surface removal [5L]
Bezier curves, B-spline curves, Hidden Surface Removal: Z-buffer algorithm, Back face detection, BSP tree method, Painter's algorithm

Text Books:
1. Hearn and P. M. Baker: Computer Graphics, 2nd ed. Prentice Hall of India, New Delhi, 1997.
2. W. M. Newman and R. F. Sproull: Principles of Interactive Computer Graphics, McGraw Hill, New Delhi, 1979.

Reference Books:
1. F. S. Hill: Computer Graphics, McMillan, New York, 1990.
2. D. P. Mukherjee: Fundamentals of Computer Graphics and Multimedia, Prentice Hall of India, New Delhi, 1999.
3. J. D. Foley et al.: Computer Graphics, 2nd ed., Addison-Wesley, Reading, Mass., 1993.
4. W. K. Giloi: Interactive Computer Graphics: Data Structure, Algorithms, Languages, Prentice Hall, Englewood Cliffs, 1978.

CO-PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS503C.1	3	1	3	3	2	-	-	-	-	-	3	3	3	3
CS503C.2	3	3	3	2	1	-	-	-	-	-	2	2	3	2
CS503C.3	-	2	2	2	2	-	-	-	-	-	1	3	3	3
CS503C.4	2	3	1	3	1	-	-	-	-	-	1	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

COURSE NAME: DATA HANDLING AND VISUALIZATION
COURSE CODE: CS503D
CONTACT (PERIODS/WEEK): 3:0:0
CREDIT POINT: 3
NO. OF LECTURES: 36

PREREQUISITES: Basics knowledge on Mathematics and Programming Skills.

COURSE OBJECTIVE(S): This course introduces students to data analysis and visualization in the field of exploration data science using Python.

COURSE OUTCOME(S): After completion of the course students will be able to

CO1: To Use data analysis tools in the Pandas library.
CO2: To Load, clean, transform, merge, and reshape data.
CO3: To Create informative visualization and summarize data sets.
CO4: To Analyze and manipulate time series data.
CO5: To Solve real-world data analysis problems.

COURSE CONTENT:

Module I: Introduction to Data Science and Python Ecosystem (6L)

- **What is Data Science**
  - Types of Data: Structured, Unstructured, Semi-structured; Data Formats: CSV, Excel, JSON, XML, SQL
  - Role of Data Handling and Visualization in Data Science

Introduction to Exploratory Data Analysis (EDA)

- Objectives and Workflow; Descriptive Statistics: Mean, Median, Mode, Variance, Skewness; Data Distribution: Histograms, Boxplots; Correlation and Covariance Analysis; Outlier Detection Techniques; Grouping and Aggregation Techniques. Python for Data Analysis:

Python for Data Analysis

- Overview and Motivation; Introduction of iPython shell and Jupyter Notebook; Essential Python Libraries: NumPy, Pandas, Matplotlib, SciPy, Scikit-learn, Statsmodels.
- Working with Jupyter Notebook and iPython

Module II: Data Handling using Pandas (8L)

- Data Structures in Pandas: Series and DataFrame
- Indexing, Selection, Filtering
- Data Loading: Reading CSV, Excel, JSON, and text files
- Writing Data to Files (CSV, Excel, HDF5)
- Data from Web APIs and Web Scraping Basics
- Handling Missing Data and Duplicates
- Data Transformation: Scaling, Encoding, Binning

- String Operations and Regular Expressions in Pandas

Module III: Data Wrangling and Visualization Techniques (6L)

- **Data Wrangling:**
  - Merging and Joining Datasets
  - Hierarchical Indexing
  - Pivoting and Reshaping Data

  Introduction to Visualization:
  - Principles of Effective Visualization
  - Visualization with Matplotlib and Seaborn
  - Charts: Bar, Line, Scatter, Histogram, Boxplot
  - Customizing Plots and Layouts

Module IV: Aggregation and Time Series Analysis (10L)

- **Group By Operations:**
  - Grouping, Aggregating, Filtering
  - Applying Multiple Functions
  - Pivot Tables and Cross Tabulations

  Time Series Analysis:
  - Working with DateTime Objects
  - Resampling and Frequency Conversion
  - Rolling, Expanding, and Moving Average Windows
  - Time Zone Handling and Period Arithmetic
  - Visualization of Time Series Data

Module V: Advanced Pandas and Project Work (6L)

- Categorical Data and Memory Optimization
- Advanced Method Chaining in Pandas
- MultiIndex and Panel Data
- Case Study: From Raw Data to Visualization
- Capstone Project:
  - Dataset Selection and Cleaning
  - Visual Storytelling
  - Final Presentation and Peer Review

TEXTBOOK:

1. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2nd edition, McKinney, W. O'Reilly Media.
2. Reimagining Data Visualization Using Python, Seema Acharya, Willey.
3. Hands-on Data Analysis & Visualization with Pandas, Purna Chander Rao. Kathula, BPB.
4. Data Analysis and Visualization Using Python: Analyse Data to Create Visualizations for Bi Systems, Dr. Ossama Embarak, Apress.

5. Data Visualization Exploring and Explaining with Data, Camm, Cochran, Fry, Ohlmann, Pub: Cengage.

REFERENCE BOOKS

1. Doing Data Science: Straight Talk from the Frontline, O'Neil, C., & Schutt, R., O'Reilly Media.
2. Learn and Practice Data Visualization using Python, Swapnil Saurav, Eka Publisher.
3. R cookbook, Teetor, P., O'Reilly. ISBN 9780596809157.
4. R graphics cookbook. Chang, W., O'Reilly. ISBN 9781449316952.
5. Discovering Statistics Using R., Andy Field, Jeremy Miles and Zoe Field., SAGE Publications Ltd. ISBN-13: 978-1446200469.
6. An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, Springer.

CO–PO MAPPING:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	2	-	-	-	-	-	-	-	3	3	3
CO2	2	3	3	2	-	-	-	-	-	-	-	3	3	3
CO3	3	3	3	2	-	-	-	-	-	-	-	3	3	3
CO4	2	2	3	3	-	-	-	-	-	-	-	3	3	3
CO5	3	3	2	3	-	-	-	-	-	-	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Soft computing
Course Code: CS504
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Discrete Mathematics, Probability and Statistics

Course Objective(s):

- Introduce students to soft computing concepts and techniques and foster their abilities in designing and implementing soft computing-based solutions for real-world and engineering problems.
- Introduce students to fuzzy systems, fuzzy logic and its applications.
- Explain the students about Artificial Neural Networks and various categories of ANN.

Course Outcome(s):

After completion of the course students will be able to

CS504.1: Understand the basic concept of soft computing and hard computing and apply them in designing solution to engineering problem.

CS504.2: Understand appropriate learning rules for each of the architectures and learn several neural network paradigms and its applications to solving engineering and other problems.

CS504.3: Apply fuzzy logic and reasoning to handle uncertainty and solving interdisciplinary engineering problems

CS504.4: Use genetic algorithms to combinatorial optimization problems and recognize the feasibility of applying a soft computing methodology for a particular problem.

CS504.5: To understand the concept and techniques of designing and implementing of soft computing methods in real world problem.

Course Content:

Module-1: Introduction to Soft Computing [8L]
An Overview of Artificial Intelligence, Evolution of Computing - Soft Computing Constituents – From Conventional Artificial Intelligence to Computational Intelligence - Machine Learning Basics.
Soft Computing: Introduction of soft computing, soft computing vs. hard computing, various types of soft computing techniques, applications of soft computing

Module-2: Fuzzy sets and Fuzzy logic[7L]
Introduction, Fuzzy sets versus crisp sets, operations on fuzzy sets, Extension principle, Fuzzy relations and relation equations, Fuzzy numbers, Linguistic variables,
Fuzzy logic, Linguistic hedges, Applications, fuzzy controllers, fuzzy pattern recognition, fuzzy image processing, fuzzy database.

Module -3: Artificial Neural Networks [9L]
Artificial Neural Network: Introduction, basic models, Hebb's learning, Adeline, Perception, Multilayer feed forward network.
Back propagation, Different issues regarding convergence of Multilayer Perceptron, Competitive learning, Self-Organizing Feature Maps, Adaptive Resonance Theory, Associative Memories, Applications.

Module -4: Genetic Algorithms [7L]
Evolutionary and Stochastic techniques: Genetic Algorithm (GA), different operators of Genetic Algorithm, Analysis of selection operations, Hypothesis of building Blocks, Schema theorem and convergence of Genetic Algorithm, Simulated annealing and Stochastic models, Boltzmann Machine, Applications.
Rough Set: Introduction, Imprecise Categories Approximations and Rough Sets, Reduction of Knowledge, Decision Tables, and Applications.

Module -5: Hybrid Systems [5L]
Neural-Network-Based Fuzzy Systems, Fuzzy Logic-Based Neural Networks, Genetic Algorithm for Neural Network Design and Learning, Fuzzy Logic controlled Genetic Algorithm. Fuzzy Logic and Genetic Algorithm for Optimization, Applications.

Text book:
1. "Neural Networks, Fuzzy logic, and Genetic Algorithms", S. Rajasekaran & G. A. V. Pai , PHI.

2. "Principles of Soft Computing", S.N. Sivanandam, S.N Deepa, Wiley publications.

3. "Neural Networks", S. Haykin, Pearson Education, 2ed, 2001.

4. "An Introduction to Genetic Algorithm", Mitchell Melanie, Prentice Hall, 1998.

Reference Books:
1. "Genetic Algorithms in Search, Optimization and Machine Learning", David E. Goldberg, Addison Wesley, 1997.

2."Intelligent Hybrid Systems", D. Ruan, Kluwer Academic Publisher, 1997.

CO – PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS504.1	3	3	3	3	-	-	-	-	-	-	-	1	-	-
CS504.2	3	3	3	3	2	-	-	-	-	-	-	2	1	-
CS504.3	3	3	3	3	2	-	-	-	-	-	-	2	-	-
CS504.4	3	3	3	3	2	-	-	-	-	-	-	-	-	-
CS504.5	3	3	3	3	-	-	-	-	2	2	-	-	-	-

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Paper name: Project Management and Finance
Code: HU(CS)501
Contact Hour-02 (L=02; T:0 P :0)
Number of Lectures:28

Course Objectives

1. To introduce students to the fundamental concepts and components of Project Management.
2. To develop the ability to perform preliminary project screening and appraisal, enabling students to identify viable project opportunities and assess their potential.
3. To provide knowledge and analytical skills for conducting comprehensive feasibility studies.
4. To impart foundational knowledge of Financial Management principles.
5. To enhance decision-making abilities related to financial management, particularly in areas such as investment analysis, cost control, and project financing.

Course Outcomes:

CO1: Understand and explain the fundamental principles, tools, and techniques of project management including planning, scheduling, monitoring, and control in engineering projects.
CO2: Apply project screening and feasibility analysis methods to assess the technical, market, and operational viability of engineering projects.
CO3: Analyze financial data to evaluate project investments, including concepts such as time value of money, break-even analysis, and risk-return trade-off.
CO4: Demonstrate decision-making capabilities in project financing and resource allocation, using basic financial management principles and tools.

Course Content:

UNIT I: BASICS OF PROJECT MANAGEMENT:

Meaning, Definition and scope and Need for Project Management - The Project Life Cycle - Phases of Project Management Life Cycle - Project Management Processes. (2L)

UNIT II: PROJECT IDENTIFICATION AND SELECTION:

Preliminary Screening of Projects. Project Identification Process- Sources of Financial resources - Pre-Feasibility Study - Feasibility Studies: Market Feasibility, Financial Feasibility and Technical Feasibility (3L)

UNIT III: PROJECT ORGANIZATION AND PLANNING:

Project manager, Cross-functional team, Dedicated project organization, Influence project organization, Matrix organization, Advantages and disadvantages of project organizations, Selection of project organization, Work Breakdown Structure (WBS), Integration of project organization and WBS, WBS and responsibility matrix. (3L)

UNIT IV: PROJECT SCHEDULING AND RESOURCE MANAGEMENT:
Gant chart, Milestone chart, Network techniques: PERT and CPM, AON and AOA representation. (4L)

UNIT-V: NATURE AND SCOPE OF FINANCIAL MANAGEMENT:
Role of financial management in business decision, the Firm and its Environment: Forms of business ownership. (2L)

UNIT-VI: BALANCE SHEET AND PROFIT AND LOSS STATEMENTS
Tools of Financial Analysis: Funds flow analysis - sources and uses of funds, measurements of cash flow, Revenue costs. (3L)
Investment Management: Capital Budgeting Techniques. PBP, ARR, Time Value of Money, NPV v/s IRR. Risk Analysis. (3L)

UNIT-VII: PROFIT RELATIONSHIPS
Break even analysis, ratio analysis, of operating and financial leverages, Working Capital Management, Credit Policy. (3L)
Financial Decision Making: Sources of raising capital, Internal financing, Cost of capital, Balanced Capital Structure. Capital Structure Theories, Dividend Policy & its Theories. (5L)

Textbooks:

1. R. Paneerselvam, P. Senthil Kumar, Project Management, PHI.
2. S. N. Maheshwari, Financial Management: Principles andApplications , Sultan Chand & Sons

Reference Books:

1. Prasanna Chandra, Projects, Planning, Analysis, Selection, Financing, Implementation and Review, Tata McGraw Hill Pvt. Ltd., New Delhi.
2. K. Nagrajan, Project Management, New Age International Publishers,
3. Vasanth Desai, Project Management, Himalaya Publications.
4. Clifford F. Gray, Erik W. Larson, Project Management, the Managerial Emphasis, Tata McGraw Hill.
6. 7. M.Y. Khan and P. K. Jain, Financial Management: Text, Problems and Cases, Tata McGraw Hill Pvt. Ltd., New Delhi.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Software Engineering Lab
Course Code: CS 591
Contact: 3:0:0
Total Contact Hours: 36
Credits: 1.5

Prerequisites:

Programming for Problem Solving

Course Objectives:

- **Apply Software Development Life Cycle (SDLC) in Practice:** To provide hands-on experience in implementing the various phases of the SDLC—requirements gathering, design, development, testing, and maintenance—using real-world scenarios or case studies.
- **Develop and Document Software Requirements:** To train students in creating Software Requirements Specifications (SRS) and related documentation through practical tools and techniques.
- **Design and Model Software Systems:** To enable students to apply design principles and model software systems using UML diagrams such as use case, class, activity, and sequence diagrams.
- **Implement and Test Software Projects:** To build working software systems based on predefined requirements and apply testing techniques to validate the correctness and quality of the developed systems.
- **Use Software Engineering Tools:** To familiarize students with essential software engineering tools such as version control (e.g., Git), project management (e.g., Trello or JIRA), CASE tools, and testing frameworks.

Course Outcomes
On completion of the course students will be able

CS591.1: To identify software development models.
CS591.2: To prepare SRS document, design document, project management related document.
CS591.3: To validate function oriented and object-oriented software design using tools.
CS591.4: To adapt various testing techniques through test cases

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content:

Module-1: [6L]

Preparation of requirement document for standard application problems in standard format. (e.g.,Library Management System, Railway Reservation system, Hospital management System, University Admission system). DFD of standard application problems.

Module-2: [6L]

Software Requirement Analysis: Describe the individual Phases/ modules of the project, Identify deliverables. Compute Process and Product Metrics (e.g. Defect Density, Defect Age, Productivity, Cost etc.) Estimation of project size using Function Point(FP) for calculation. Cost Estimation models. L

Module-3: [6L]

Use Case diagram, Class Diagram, Sequence Diagram, Activity Diagram and prepare Software Design Document using tools like Rational Rose.(For standard application problems)

Module-4:[9L]

Software Development, Coding Practice and Debugging, Design Test Script/Test Plan(both Black box and White Box approach)

Module-5:[9L]

Software project management, Project planning and control, configuration control, cost estimation, project scheduling using PERT and GANTT charts, cost-time relations using standard tools.

Text Books:

1. Fundamentals of Software Engineering by Rajib Mall, –PHI-3rd Edition, 2009.
2. Software Engineering-Pankaj Jalote(Wiley-India)

Reference Books:

1. Software Engineering–Agarwal and Agarwal(PHI)
2. Software Engineering, by Ian Sommerville, Pearson Education Inc., New Delhi, (2009).
3. Software Engineering: A Practitioner‟s Approach”, by Roger S. Pressman, McGraw-Hill.(2005)

CO–PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	3	3	3	3	2	1	1	-	3	2	2	2	3	2
CO2	3	3	2	3	3	1	1	-	3	3	2	2	3	2
CO3	2	3	2	3	3	-	2	-	3	2	2	3	3	3
CO4	2	3	2	3	2	-	1	-	3	2	2	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Object Oriented Programming using Java Lab
Course Code: CS592
Contact (Periods/Week): 0:0:3
Credit Point: 1.5
No. of Lectures: 36

Prerequisites:
1. Computer Fundamentals
2. Basic understanding of Computer Programming and related Programming Paradigms
3. Problem Solving Techniques with proper logic Implementation

Course Objective(s):
- It demonstrates that how can you change the implementation of an object without affecting any other code by increasing data security and protecting unwanted data access. (Encapsulation).
- It allows you to have many different functions, all with the same name, all doing the same job, but depending upon different data. (Polymorphism).
- It guides you to write generic code: which will work with a range of data, so you don't have to write basic stuff over, and over again. (Generics).
- It lets you write a set of functions, then expand them in different direction without changing or copying them in any way. (Inheritance)

Course Outcome(s):
After completion of the course students will be able to

CO1: To describe the object-oriented approach in Java by outlining the relationship between classes, objects, and constructors.
CO2: To apply the concept of code reusability in Java through the use of inheritance and class hierarchies.
CO3: To implement Java programs using encapsulation, polymorphism, and relevant object-oriented keywords.
CO4: To analyze object-oriented features such as data abstraction, packages, and interfaces to determine their roles in Java programming.
CO5: To construct Java applications using exception handling, multithreading, and applet-based web programming techniques.

Course Content:
1. Basic Java programs such as
    i. printing “Hello, GNIT”.
    ii. checking whether a number is even or odd.
    iii. finding out the roots of a quadratic equation.
    iv. finding out the factorial of a given number.
    v. printing Fibonacci series up to n terms.

vi. Creating a class calculator that has 4 methods like add, sub, mul & div. Then doing the addition, subtraction, multiplication, and division of 2 integer numbers using these 4 methods.

2. Java programs to implement default constructor, parameterized constructor using command line argument, 'this' keyword.

3. Java programs to implement method overloading, constructor overloading, call by value, call by reference, recursion.

4. Java programs to implement the difference between public and private access specifier, 'static' keyword, inner class.

5. Java programs to implement simple inheritance, hierarchical inheritance, and multilevel inheritance.

6. Java programs to implement 'super' keyword to access a superclass member, 'super' keyword to access a superclass constructor, method overriding.

7. Java programs to implement run-time polymorphism, abstract class and method.

8. Java programs to implement interface, multiple inheritance.

9. Java programs to
   i. create two user-defined packages pkg1 and pkg2 and import both to another program which is outside the packages.
   ii. create multiple packages containing classes with identical names.
   iii. show how a protected variable of one package can be accessed in a subclass in another package.
   iv. show how to add multiple public classes to a single package.

10. Java programs to implement Arithmetic Exception, Array Index Out of Bounds Exception, 'throw' and 'throws' keywords, finally block.

11. Java programs to
   i. create 3 threads - the 1st thread to display GOOD MORNING for every 1 second, the 2nd thread to display HELLO for every 2 seconds and the 3rd thread to display WELCOME for every 3 seconds.
   ii. implement the above program by assigning priorities to the created threads such that the 1st thread executes first followed by the 2nd thread and lastly the 3rd thread.

12. Java programs to
   i. develop an applet that display simple message.
   ii. develop an applet that will add two integer numbers.

iii. develop an applet that will draw lines, rectangle and oval.

Text book:

1. Herbert Schildt – "Java: The Complete Reference " – 9th Ed. – TMH
2. E. Balagurusamy – " Programming with Java: A Primer " – 3rd Ed. – TMH.

Reference

1. R. K. Das – " Core Java for Beginners " – VIKAS PUBLISHING.
2. Rambaugh, James Michael, Blaha – " Object Oriented Modelling and Design " – Prentice Hall, India

CO-PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	3	3	2	-	-	-	-	-	3	2	2	3
CO2	1	3	2	2	3	-	-	-	-	-	1	2	2	3
CO3	2	-	1	2	2	-	-	-	-	-	2	2	2	3
CO4	3	2	3	2	3	-	-	-	-	-	2	2	2	3
CO5	2	3	1	3	2	-	-	-	-	-	1	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Compiler Design Lab
Course Code: CS593A
Contact: 0:0:3
Credits: 1.5

Course Objective(s):
To implement Lexical Analyzer using Lex tool & Syntax Analyzer or parser using YACC Tool
To implement NFA and DFA from a given regular expression
To implement front end of the compiler by means of generating Intermediate codes
To implement code optimization techniques

Course Outcomes:
On successful completion of the learning sessions of the course, the learner will be able
CO1: Design & conduct experiments for NFA and DFA from a given regular expression.
CO2: Apply the knowledge of lex tool & yacc tool to develop a scanner & parser.
CO3: Design & implement a front end of the compiler.
CO4: Develop program for implementing symbol table.
CO5: Develop program for solving parser problems

Module 1:
NFA Construction from a regular expression, Conversion between NFA and DFA.

Module 2:
Use LEX tool to implement alexical analyzer, Use YACC tool to implement a syntax analyser or parser.

Module 3:
Implementation of a recursive descent parser for an expression grammar that generates arithmetic expressions with digits, + and*, Checking whether a string belongs to a grammar or not.

Module 4:
Calculation of leading & trailing for all the non-terminals of the given grammar, Calculation of FIRST, FOLLOW of the given grammar.

Module 5:
Identifying whether a given string is an identifier or not, identifying whether a string is a key word or not, identifying whether a string is a constant or not.

Recommended Books
1. Das,V.V. (2007). Compiler Design using FLEX and YACC. PHI Learning Pvt. Ltd.
2. Mason, T., & Brown, D. (1990). Lex & yacc. O'Reilly & Associates, Inc.
3.Johnson, S.C. (1975). Yacc: Yet another compiler compiler (Vol.32). Murray Hill, NJ: Bell Laboratories.

CO–PO-PSO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	-	3	-	-	2	1	-	-	1	-	-	1	2
CO2	-	1	2	2	-	-	1	-	1	-	-	1	2	-
CO3	2	-	2	-	1	-	2	1	-	-	1	-	2	3
CO4	1	2	-	2	2	1	-	-	2	-	-	2	-	2
CO5	2	3	1	1	-	1	-	2	-	-	-	2	-	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Cryptography and Network Security LAB
Course Code: CS593B
Contact (Periods/Week): 0:0:3
Credit Point: 1.5
No. of Lectures: 36

Prerequisites:

1. Knowledge of Computer Networks and Operating Systems fundamentals
2. Understanding of Discrete Mathematics concepts

Course Objective(s):

- To impart concepts on cryptography and Network security
- To gain knowledge of the standard algorithms used to provide confidentiality, integrity, and authenticity
- To recognize the various key distribution and management systems for security of a cryptosystem

Course Outcomes:

After completion of course, students would be able

CS593B.1: To understand fundamental network security concepts, including services, classical encryption, and number theory, to analyze basic cryptographic mechanisms.

CS593B.2: To apply symmetric and asymmetric encryption algorithms, such as DES, AES, RSA, and elliptic curve cryptography, to design secure communication systems and manage cryptographic keys.

CS593B.3: To analyze and evaluate authentication mechanisms, hash functions, and digital signature schemes to ensure integrity, authenticity, and non-repudiation in data communication.

CS593B.4: To design and assess secure network architectures using firewalls, intrusion detection systems, and trusted systems, and implement practical countermeasures against various threats and attacks.

CS593B.5: To integrate and apply security protocols for email, IP, and web applications (e.g., PGP, S/MIME, IPSec, SSL/TLS) to achieve end-to-end secure communications in real-world scenarios.

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content:

Module I: Introduction to Cryptography Tools [6]

- Introduction to cryptography and network security basics.
- Installation & setup of tools:
  - **OpenSSL**
  - **Kali Linux**
  - **Wireshark**
  - **CrypTool**
- Study of different types of attacks: passive vs. active

Sample Experiments:

- Use Wireshark to capture and analyze packets.
- Basic encryption/decryption using OpenSSL.

Module II: Classical Encryption Techniques [6]

- Caesar Cipher
- Playfair Cipher
- Hill Cipher
- Vigenère Cipher
- Mono-alphabetic and Poly-alphabetic ciphers

Sample Experiments:

- Implement Caesar, Playfair, and Hill ciphers in C/Python.
- Frequency analysis for breaking substitution ciphers.

Module III: Symmetric Key Cryptography [6]

- Block and Stream Ciphers
- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)
- RC4 Algorithm

Sample Experiments:

- Implement DES/AES encryption and decryption in Python or Java.
- Study modes of operation: ECB, CBC, CFB, OFB.

Module IV: Asymmetric Key Cryptography [6]

- RSA Algorithm
- Diffie-Hellman Key Exchange
- ECC (Elliptic Curve Cryptography) – Basic understanding

Sample Experiments:

- Implement RSA encryption/decryption.

- Demonstrate key exchange using Diffie-Hellman.
- Use OpenSSL to generate public-private keys and encrypt/decrypt messages.

Module V: Hashing and Digital Signatures [6]

- MD5, SHA-1, SHA-256 Hashing algorithms
- HMAC (Hash-based Message Authentication Code)
- Digital Signature using RSA

Sample Experiments:

- Generate message digests using MD5/SHA-1.
- Implement HMAC for secure communication.
- Simulate digital signature creation and verification.

Module VI: Network Security Protocols & Tools [6]

- SSL/TLS – Concepts and demo with OpenSSL
- Firewall Configuration (iptables/ufw)
- Intrusion Detection Systems (Snort)
- VPN and Secure Shell (SSH)

Sample Experiments:

- Configure and test SSH for secure communication.
- Analyze HTTPS handshakes using Wireshark.
- Use ip tables to block/unblock specific ports/IPs.

Text books:

1. Kahate, A. (2013). Cryptography and network security. Tata McGraw-Hill Education.
2. Forouzan, B. A., & Mukhopadhyay, D. (2015). Cryptography and network security. New York, NY: McGraw Hill Education (India) Private Limited.

Reference books:

1. Stallings, W. (2006). Cryptography and network security, 4/E. Pearson Education India.
2. Daras, N. J., & Rassias, M. T. (Eds.). (2015). Computation, cryptography, and network security (pp. 253-287). Springer.
3. Kumar, A., & Bose, S. (2017). Cryptography and network security. Pearson Education India.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	2	1	1	-	-	-	1	2	1	2	1	1
CO2	2	2	3	2	3	-	-	-	1	1	1	2	2	1
CO3	1	2	2	3	2	-	-	-	2	3	1	2	2	2
CO4	2	2	3	2	3	-	-	-	1	1	1	2	1	2
CO5	2	1	2	1	2	-	-	-	1	1	2	2	1	1

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Computer Graphics Lab
Course Code: CS593C
Contact: 0:0:3
Total Contact Hours: 36
Credits: 1.5

Prerequisites:
Mathematics, Knowledge of C programming language.

Course Objectives:

- To make students aware of the concepts underlying modern **Computer Graphics** and **Machine Vision**.
- To enable students to understand and apply **digital image synthesis techniques**.
- To develop generic skills for **designing algorithms** in computer graphics.
- To prepare students to solve **broad-based computing problems** using graphics and vision techniques.
- To apply these skills across **various domains** requiring computer graphics and machine vision.

Course Outcomes:
On successful completion of the learning sessions of the course, the learner will be able
CO1**:** To remember the Geometric primitives.
CO2: To analyze the concept of scan line polygon filling.
CO3: To explain basic transformations on objects.
CO4: To experiment all algorithm on 2D space.

Course Contents:
Module I: Introduction to Graphics Programming (2 Lab Sessions)
Study and understand basic graphics functions defined in graphics.h.
Draw different graphical objects (line, circle, rectangle, polygon) using functions in graphics using C.

Module II: Graphical Objects **Drawing Algorithms (3 Lab Sessions)**
Implement program of Line Drawing using DDA Algorithm.
Implement program of Line Drawing using Bresenham's Algorithm.
Implement program of Circle Drawing using Bresenham's Algorithm.
Implement program of Ellipse Drawing using Bresenham's Algorithm.

Module III: Transformations (4 Lab Sessions)
Programs to perform 2D translation, rotation, scaling, reflection, and shearing on different graphical objects.

Programs to perform 3D translation, rotation, scaling, reflection, and shearing on different graphical objects

Module IV: Polygon Filling Algorithms (1 Lab Sessions)
Implement program of Flood-Fill Algorithm for polygon filling.
Implement program of Boundary-Fill Algorithm for polygon filling.
Implement program of Scan Line Polygon Filling Algorithm.

Module V: Clipping Algorithms (2 Lab Sessions)
Programs to study window to viewport transformations

Program for Cohen Sutherland Line clipping algorithm.

Text Books:

1. Hearn and P. M. Baker: Computer Graphics, 2nd ed. Prentice Hall of India, New Delhi, 1997.
2. W. M. Newman and R. F. Sproull: Principles of Interactive Computer Graphics, McGraw Hill, New Delhi, 1979.

Reference Books:

1. F. S. Hill: Computer Graphics, McMillan, New York, 1990.
2. D. P. Mukherjee: Fundamentals of Computer Graphics and Multimedia, Prentice Hall of India, New Delhi, 1999.
3. Schaum's outlines Computer Graphics (2nd Ed.) by Ray A. Plastock, Gordon Kalley, McGraw-Hill Inc.
4. 2. Mathematical Elements for Computer Graphics by David Rogers, J. Alan Adams, McGraw Hill Education

CO-PO-PSO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CS593C.1	3	1	3	3	2	-	-	-	-	-	3	3	3	3	3
CS593C.2	3	3	3	2	1	-	-	-	-	-	2	1	2	3	2
CS593C.3	-	2	2	2	2	-	-	-	-	-	1	3	3	3	3
CS593C.4	2	3	1	3	1	-	-	-	-	-	1	3	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

COURSE NAME: DATA HANDLING AND VISUALIZATION LAB

COURSE CODE: CS593D

CONTACT (PERIODS/WEEK): 0:0:3

CREDIT POINT: 1.5

PREREQUISITES: Basic programming knowledge (preferably Python), familiarity with statistics, data types, and file handling, along with analytical thinking and general computer literacy are required prerequisites**.**

COURSE OBJECTIVE(S): To equip students with practical skills in data handling, analysis, and visualization using Python tools for effective data-driven decision making**.**

COURSE OUTCOME(S): After completion of the course students will be able to

CO1: To understand the Python ecosystem and use essential libraries for data analysis and visualization.

CO2: To perform data cleaning, transformation, and exploratory data analysis using Pandas and statistical techniques.

CO3: To apply data wrangling methods such as merging, reshaping, and indexing to prepare datasets for analysis.

CO4: To create effective visualizations using Matplotlib and Seaborn to interpret and communicate data insights.

CO5: To develop and present a complete data analysis project from raw data to visual storytelling using appropriate tools and techniques.

COURSE CONTENT:

Week 1: Introduction to Python and the Data Science Ecosystem

- Setting up Python environment (Anaconda/Jupyter); Introduction to Jupyter Notebook and iPython; Basic Python syntax: variables, data types, functions; Overview of libraries: NumPy, Pandas, Matplotlib, Seaborn
  Lab Activities: Create a notebook, execute basic Python scripts, Import and use NumPy for simple array operations.

Week 2: Exploratory Data Analysis (EDA) – Part 1

- Descriptive statistics: mean, median, mode, variance, skewness; Introduction to pandas Series and DataFrames; Loading datasets (CSV, Excel)

Lab Activities: Load real datasets and compute descriptive stats, Visualize distributions using histograms and boxplots

Week 3: Exploratory Data Analysis (EDA) – Part 2

- Correlation and covariance; Outlier detection and handling; Grouping and aggregation

  Lab Activities: Identify correlations; Handle outliers using IQR and Z-score methods; Group and aggregate data by categories

Week 4: Data Handling with Pandas – Part 1

- Data selection, filtering, indexing; Handling missing and duplicate values; String operations and regex in Pandas.

  Lab Activities: Practice with .loc[], .iloc[], .isin(), etc.; Clean data using .dropna(), .fillna(), .duplicated()

Week 5: Data Handling with Pandas – Part 2

- Data transformation: scaling, encoding, binning; Data loading from JSON, text, APIs (intro); Writing data to various formats (CSV, Excel, HDF5).
  Lab Activities: Use LabelEncoder, MinMaxScaler; Fetch simple data from public APIs using requests; Export cleaned data to various formats.

Week 6: Data Wrangling Techniques

- Merging and joining datasets; Hierarchical indexing; Pivoting and reshaping data.
  Lab Activities: Merge sales and customer datasets; Practice pivot_table() and melt().

Week 7: Introduction to Data Visualization

- Principles of effective data visualization; Visualization using Matplotlib and Seaborn.
  Lab Activities: Generate basic bar, line, scatter plots; Customize plot aesthetics (labels, titles, legends).

Week 8: Advanced Plotting and Layout Customization

- Boxplot, violin plots, heatmaps; Subplots, plot grids, and color palettes.
  Lab Activities: Create comparative plots using subplots; Customize plots for publication-style output.

Week 9: Group By and Aggregation

- Group By: splitting, applying, combining; Applying multiple functions; Pivot tables and cross-tabulations.
  Lab Activities: Perform grouped stats (e.g., avg sales by region); Create pivot tables and interpret results.

Week 10: Time Series Analysis

- Working with Date Time objects; Resembling and frequency conversion; Moving averages and time zone adjustments.
  Lab Activities: Import time-indexed data (e.g., stock prices, weather); Apply rolling windows and visualize time-based trends.

Week 11: Advanced Pandas

- Categorical data and memory optimization; Method chaining and Multi Index; Introduction to panel data.
  Lab Activities: Convert columns to categorical; Chain Pandas methods for efficient processing.

Week 12: Capstone Mini Project & Revision

- Final Project Work: Select dataset, Clean and transform data, Visualize findings, Create a narrative; Peer review and feedback

  Lab Activities: Final Jupyter notebook, Presentation (5–10 mins), Evaluation based on EDA, visualization, storytelling, and reproducibility.

CO–PO MAPPING:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO2	3	2	3	3	-	-	-	-	-	-	-	3	3	3
CO3	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO4	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO5	3	3	2	3	-	-	-	-	-	-	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Soft computing Lab
Course Code: CS594
Contact (Periods/Week): 0:0:3
Credit Point: 1.5

Prerequisites: Discrete Mathematics, Probability and Statistics

Course Objective(s):

- Introduce students to soft computing concepts and techniques and foster their abilities in designing and implementing soft computing-based solutions for real-world and engineering problems.
- Introduce students to fuzzy systems, fuzzy logic and its applications.
- Explain the students about Artificial Neural Networks and various categories of ANN.

Course Outcome(s):

After completion of the course students will be able to

CS594.1: Understand the basic concept of soft computing and hard computing and apply them in designing solution to engineering problem.
CS594.2: Understand appropriate learning rules for each of the architectures and learn several neural network paradigms and its applications to solving engineering and other problems.
CS594.3: Apply fuzzy logic and reasoning to handle uncertainty and solving interdisciplinary engineering problems
CS594.4: Use genetic algorithms to combinatorial optimization problems and recognize the feasibility of applying a soft computing methodology for a particular problem.
CS594.5: To understand the concept and techniques of designing and implementing of soft computing methods in real world problem.

Course Content:

1. Program in python to perform Union, Intersection and Complement operations of Fuzzy set.
2. Program in python to implement De Morgan's law.
3. Program in python to plot various membership functions.
4. Program in python to impalement Fuzzy inference system.
5. Program in python to generate XOR function using back propagation algorithm.
6. Program in python for solving linearly separable problem using Perceptron Model
7. Program in python for maximizing a function using Genetic Algorithm.
8. Program in python for implementing two input sine function.
9. Program in python for implementing three input non-linear function.

Text book:

1.“Neural Networks, Fuzzy logic, and Genetic Algorithms”, S. Rajasekaran & G. A. V. Pai , PHI.
2.“Principles of Soft Computing”, S.N.Sivanandam, S.N Deepa, wiley publications.
3.“Neural Networks”, S. Haykin, Pearson Education, 2ed, 2001.
4.“An Introduction to Genetic Algorithm”, Mitchell Melanie, Prentice Hall, 1998.

Reference Books:

1. “Genetic Algorithms in Search, Optimization and Machine Learning”, David E. Goldberg, Addison Wesley, 1997.
2.“Intelligent Hybrid Systems”, D. Ruan, Kluwer Academic Publisher, 1997.

CO – PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS594.1	3	3	3	3	-	-	-	-	-	-	-	1	-	-
CS594.2	3	3	3	3	2	-	-	-	-	-	-	2	1	-
CS594.3	3	3	3	3	2	-	-	-	-	-	-	2	-	-
CS594.4	3	3	3	3	2	-	-	-	-	-	-	-	-	-
CS594.5	3	3	3	3	-	-	-	-	2	2	-	-	-	-

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

3rd Year 6th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS601	Web and Internet Technology	3	0	0	3	3
2	ENGG	Major	CS602	Deep Learning	3	0	0	3	3
3	ENGG	Major	CS603A	Image Processing	3	0	0	3	3
			CS603B	Cloud Computing					
			CS603C	Big Data and Data Analytics					
			CS603D	Natural Language Processing					
4	ENGG	Major	CS604A	Mobile Computing	3	0	0	3	3
			CS604B	Human Computer Interaction					
			CS604C	E-Commerce and Digital Business Model					
			CS604D	Quantum Computing					
5	ENGG	Minor	CS605	Cyber Law and Ethics	3	0	0	3	3
B. PRACTICAL									
1	ENGG	Major	CS691	Web and Internet Technology Lab	0	0	3	3	1.5
2	ENGG	Major	CS692	Deep Learning Lab	0	0	3	3	1.5
3	ENGG	Major	CS693A	Image Processing Lab	0	0	3	3	1.5
			CS693B	Cloud Computing Lab					
			CS693C	Data Analytics Lab					
			CS693D	Natural Language Processing Lab					
4	PRJ	Project	CS681	Project-II	0	0	8	8	4
Total of Theory, Practical								**32**	**23.5**
Total Third Year Credit									**45.5**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Web and Internet Technology
Course Code: CS601
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Computer Networks

Course Objective(s):

- To impart the design, development and implementation of Static and Dynamic Web Pages.
- To develop programs for Web using Scripting Languages
- To give an overview of Server-Side Programming in Web.

Course Outcome(s):
After completion of the course students will be able to

CO1: To define the concepts of World Wide Web (www), Internet, HTTP Protocol, Web Browsers, Client-Server etc.
CO2: To summarize interactive web pages using HTML, DHTML and CSS.
CO3: To implement the knowledge of different information interchange formats like XML.
CO4: To explain web applications using scripting languages like JavaScript, CGI, PHP.
CO5: To write different server-side programming like Servlet, JSP.

Course Content:

Module 1-Introduction to Web and Internet Technology [5L]:
Concept of World Wide Web (www), Internet and the relation with www [1L]; The Internet - Basic Internet Protocols, HTTP Protocol - Request and Response, Web browser [1L]; Web clients and Web servers, Dynamic IP [1L]; Clients, Servers, and Communication, Web site design principles, Planning the site and navigation [2L].

Module -2: Web Designing [12L]
HTML, DHTML & CSS [8L]: Introduction, Elements, Attributes, Heading, Paragraph. Formatting [2L]; Link, Table, List, Block, Layout, Html Forms and input [1L]; IFrame, Colors, Image Maps and attributes of image area [2L]; Introduction to CSS, basic syntax and structure of CSS, different types- internal, external and inline CSS [1L]; Basic Introduction of DHTML, Difference between HTML and DHTML, Documentary Object Model (DOM) [2L].

Extended Markup Language (XML) [4L]: Introduction, Difference between HTML & XML, XMLTree [2L]; Syntax, Elements, Attributes, Validation and parsing, DTD [2L].

Module 3- Web Scripting [9L]
Java Scripts [4L]: Basic Introduction, Statements, comments, variable, operators, data types[1L]; condition, switch, loop, break [1L]; Java script functions, objects, and events[2L].

CGI Scripts [1L]: Introduction, Environment Variable, GET and POST Methods.

PHP Scripting [4L]: Introduction, Syntax, Variables, Output, Data types, String, Constants [1L]; Operator, Decision Control statements [1L]; switch-case, Loop, PHP function[1L]; array, Form Handling[1L].

Module 4 - JSP and Servlet [11L]:
Java Server Page (JSP) [8L]: JSP Architecture [1L]; JSP Servers, JSP Life Cycle [1L]; Understanding the layout of JSP, JSP Script-let Tag [1L]; JSP implicit object (request and response) [1L]; Variable declaration, methods in JSP [1L]; JSP directive (Tag-lib and include), JavaBean- inserting JavaBean in JSP [1L]; JSP Action tags (Forward & Include) [1L]; Creating ODBC data source name, Introduction to JDBC, prepared statement and callable statement [1L].

Java Servlet [3L]: Servlet environment and role, Servlet life cycle [1L]; Servlet methods-Request, Response, Get and post [1L]; Cookies and Session [1L].

Text book:

1. “Web Technology: A Developer's Perspective”, N.P. Gopalan and J. Akilandeswari, PHI Learning, Delhi, 2013. (Topics covered: html, CSS, imagemap, xml)
2. “Learning PHP, MySQL & JavaScript”, Robin Nixon, O’Reilly Publication. (Topics covered: PHP, Java Script)
3. “Head First Servlet’s & JSP”, Bryan Basham, Kathy Sterra, Bert Bates, O’Reilly Publication. (Topics covered: Servlet, JSP)

Reference
1. “Programming the World Wide Web”, Robert. W. Sebesta, Fourth Edition, Pearson Education, 2007.

2. “Core Web Programming”- Second Edition-Volume I and II, Marty Hall and Larry Brown, Pearson Education, 2001.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	1	3	3	2	-	-	-	-	-	3	2	1	2
CO2	3	2	3	2	2	-	-	-	-	-	2	2	2	3
CO3	2	3	2	2	2	-	-	-	-	-	1	2	3	2
CO4	2	2	3	2	3	-	-	-	-	-	2	3	3	1
CO5	2	3	1	3	1	-	-	-	-	-	1	2	2	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Paper Name: Deep Learning
Paper Code: CS602
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisite: A solid background in Statistics, Calculus, Linear Algebra, and Probability. Good exposure to Python packages like NumPy, Pandas, Matplotlib, and Scikit-learn.

Course Objectives:

- To introduce the fundamental techniques and principles of Neural Networks.
- To study different models in ANN and their applications.
- To familiarize deep learning concepts with CNN, RNN, and **Autoencoders**.
- To expose students to practical implementation using modern deep learning frameworks (Keras/PyTorch).

Course Outcomes:

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

CO1: Understand the basic concepts in Neural Networks and Deep Learning and their applications.

CO2: Understand the architecture and function of Shallow & Deep Neural Networks.

CO3: Understand Convolutional Neural Network models for image-based tasks.

CO4: Understand Recurrent Neural Network models for sequence data.

CO5: Apply Autoencoders and Generative Models for feature extraction and data generation.

CO6: Implement and experiment with deep learning models using industry-standard frameworks like Keras or PyTorch.

Course Content:

Module 1 [8L]

Neural Network Fundamentals, Supervised Learning with Neural Networks, Binary Classification and Logistic Regression, Logistic Regression Cost Function, Gradient Descent Optimization, Computation Graphs and Derivatives, Vectorization in Practice, Vectorizing Logistic Regression and its Gradient

Module 2 [7L]
Shallow Neural Network Architecture, Neural Network Representation and Output Computation, Vectorizing Across Multiple Training Examples, Non-Linear Activation Functions and Their Derivatives, Gradient Descent for Neural Networks, Backpropagation Intuition and Random Initialization, Parameters vs. Hyperparameters

Module 3 [6L]
Foundations of Computer Vision, Convolutional Layer Operations (Padding, Strided Convolutions), Pooling Layers (Max, Average), Classic CNN Architectures (e.g., ResNets, Inception, MobileNet), Transfer Learning and Data Augmentation Techniques, Object Detection Algorithms (e.g., YOLO)

Module 4 [7L]
Recurrent Neural Network (RNN) Models, Backpropagation Through Time and the Vanishing Gradients Problem, Gated Recurrent Units (GRU) & Long Short-Term Memory (LSTM), Word Representation and Embeddings (Word2Vec, GloVe), Sequence-to-Sequence Architectures, Beam Search and Error Analysis, The Attention Mechanism and Transformer Networks

Module 5 [8L]
Introduction to Autoencoders, Undercomplete Autoencoders for Dimensionality Reduction, Sparse Autoencoders, Denoising Autoencoders for Noise Reduction, Practical Applications of Autoencoders, Introduction to Generative Models, Generative Adversarial Networks (GANs) Fundamentals, Practical Implementation with Keras and PyTorch

Text Books

1. Charu C. Aggarwal, "Neural Networks and Deep Learning: A Textbook", Springer; 1st ed. 2018 edition
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press
3. **Francois Chollet, "Deep Learning with Python", Manning Publications; 2nd edition** (Promoted for practical focus)

Reference Books

1. Simon Haykin, "Neural Networks and Learning Machines", Pearson Prentice Hall, 3rd Edition
2. Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jess, "Neural Network Design (2nd Edition)"

CO-PO Mapping

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CS504C.1	-	2	3	3	3	-	-	-	-	-	1	1	-	3
CS504C.2	-	2	2	2	1	-	-	-	-	-	-	1	2	3
CS504C.3	-	2	3	3	3	3	-	-	-	-	2	1	2	2
CS504C.4	3	2	2	2	1	3	2	-	-	-	3	2	-	-
CS504C.5	2	2	2	2	2	-	-	-	-	-	1	2	-	
CS504C.6	2	2	3	3	3	2	-	-	-	1	2	3	-	-

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Image Processing
Course Code: CS603A
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Fourier analysis, Linear algebra and Probability

Course Objective(s):

- To learn discrete Fourier transform and its properties
- To study the monochrome and color image fundamentals
- To learn the analytical tools and methods which are currently used in digital image processing as applied to image information for human viewing.
- To learn image compression and segmentation techniques.

Course Outcome(s):

After completion of the course students will be able to
CO1: To acquire the knowledge of basic pre-processing techniques in monochrome and color images.
CO2: To develop skill in concepts of image enhancement like linear and nonlinear spatial filters using MATLAB.
CO3: To understand the concept and techniques of simple image processing projects using different methods of restoration.
CO4: To acquire the knowledge of the various segmentation algorithms for practical applications.
CO5: To analyze the performance of Lossless and Lossy compression techniques in images.

Course Content:

Module -1: [5L]
Digital Image Fundamentals: Overview, Computer imaging systems, Digital Image Representation, Fundamental steps in Image Processing; Elements of Digital Image Processing - Image Acquisition, Storage, Processing, Communication, Display; Digital Image Formation: A Simple Image Model, Use and Analysis of Color Models in Image Processing; Sampling & Quantization - Uniform & Non-uniform.

Module -2: [5L]
Mathematical Preliminaries: Neighbor of pixels, Connectivity, Relations, Equivalence & Transitive Closure; Distance Measures, Arithmetic/Logic Operations, Discrete Signals and Systems; A Review – Fourier Transformation, Properties of The Two Dimensional Fourier Transform; Discrete Fourier Transform, Discrete Cosine & Sine Transform.

Module 3: [6L]
Image Enhancement: Spatial Domain Gray level transformations, Histogram processing; Basics of Spatial Filtering, Smoothing and Sharpening Spatial Filtering, Frequency Domain; Smoothing and Sharpening frequency domain filters- Ideal, Butterworth and Gaussian filters.

Module -4: [7L]
Image Restoration and Segmentation: mean & adaptive filters, degradation model, inverse filter; Discrete Formulation, Algebraic Approach to Restoration Unconstrained & Constrained; Constrained Least Square Restoration, Restoration by Homomorphic Filtering; Geometric Transformation: Spatial Transformation, Gray Level Interpolation. Image Segmentation: Point Detection, Line Detection, Edge detection, combined detection.

Module -5: [8L]
Edge Linking, Boundary Detection and Image compression: Edge Linking & Boundary Detection- Local Processing, Global Processing via The Hough Transform; Thresholding - Foundation, Simple Global Thresholding, Optimal Thresholding; Region Oriented Segmentation - Basic Formulation, Region Growing by Pixel Aggregation, Region Splitting & Merging, Image compression: system model, lossless methods, Lossy methods.

Module -6: [5L]
Image Representation and Recognition: Boundary representation, Chain Code, Polygonal approximation, signature, boundary segments, Boundary description, Shape number, Fourier Descriptor, moments, Regional Descriptors, Topological feature, Texture Patterns and Pattern classes, Recognition based on matching

Text Books:
1. Chanda &Majumder, Digital Image Processing & Analysis, PHI
2. Rafael C. Gonzales and Richard E. Woods, Digital Image Processing, Third Edition, Pearson Education,2010.

Reference books:
1. Malay K. Pakhira, Digital Image Processing and Pattern Recognition, First Edition, PHI
   Learning Pvt. Ltd.,2011.
2. The Image Processing Handbook, 7th Edition, by John C. Russ, F. Brent Neal , CRC Press.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	-	-	2	-	1	1	-	-	-	-	3	3	1
CO2	-	1	2	1	-	-	-	-	-	-	-	3	3	1
CO3	1	2	-	3	2	-	-	-	1	-	-	3	3	3
CO4	2	-	-	-	-	1	-	-	-	1	1	3	3	3
CO5	-	3	-	2	-	1	-	1	-	-	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Paper Name: Cloud Computing
Code: CS603B
Contacts: 3:0:0
Credits: 3
Total Contact hours: 36L

Prerequisite

1. Should have the basic knowledge of Operating Systems.
2. Should be aware of the fundamental concepts of Networking.
3. Should have knowledge of heterogeneous systems and resource management.

Course Objective(s):

- To learn the workflow of cloud business model and optimized resource allocation.
- To gain knowledge of cloud service and delivery models and develop an understanding of the risk and compliance responsibilities and Challenges for each Cloud type and service delivery model.
- To learn virtualization techniques, load balancing, and work strategy of different cloud infrastructure.
- To know the security and privacy issues in cloud infrastructure

Course Outcomes:
After completion of course, students would be able:

CS603B.1: To identify the business model concepts, architecture and infrastructure of cloud computing, including cloud service models and deployment models.
CS603B.2: To journaling some important cloud computing driven commercial systems such as Google Apps, Microsoft Azure and Amazon Web Services and other business applications
CS603B.3: To articulate and design suitable Virtualization concept, Cloud Resource Management and design scheduling algorithms.
CS603B.4: To categorize the core issues of cloud computing such as security, privacy, interoperability, and its impact on cloud application.

Course Contents:
Module 1: Definition of Cloud Computing and its Basics [8L]
Definition of Cloud Computing: Defining a Cloud, Cloud Types – NIST model, Cloud Cube model, Deployment models (Public, Private, Hybrid and Community Clouds), Service models – Infrastructure as a Service, Platform as a Service, Software as a Service with examples of services/ service providers, Cloud Reference model, Characteristics of Cloud Computing – a shift in paradigm Benefits and advantages of Cloud Computing [3]
Cloud Architecture: Cloud Infrastructure, Architecture of each components, Virtualization versus Traditional Approach, Virtualization Model for Cloud Computing. [2]
Services and Applications by Type [3]
IaaS – Basic concept, Workload, partitioning of virtual private server instances, Pods, aggregations, silos.

PaaS – Basic concept, tools and development environment with examples
SaaS - Basic concept and characteristics, Open SaaS and SOA, examples of SaaS platform
Identity as a Service (IDaaS) Compliance as a Service (CaaS)

Module 2: Use of Platforms in Cloud Computing [6L]
Concepts of Abstraction and Virtualization [2L]
Virtualization technologies: Types of virtualization, Load Balancing and Virtualization: Basic Concepts, Network resources for load balancing; Classification of Virtualization Environment: Scheduling-based Environment, Load-Distribution-Based Environment, Energy Aware-Based Environment, Operational-Based Environment, Distributed Pattern-Based Environment, Transactional-Based Environment
Mention of The Google Cloud as an example of use of load balancing Hypervisors: Virtual machine technology and types, VMware vSphere Machine imaging (including mention of Open Virtualization Format – OVF) [2L]
Porting of applications in the Cloud: The simple Cloud API and AppZero Virtual Application appliance
Concepts of Platform as a Service [2L]
Definition of services, Distinction between SaaS and PaaS (knowledge of Salesforce.com and Force.com), Application development. Use of PaaS Application frameworks.

Module 3: Cloud Service Models [6L]
Use of Google Web Services [2L]
Discussion of Google Applications Portfolio – Indexed search, Dark Web, Aggregation and disintermediation, Productivity applications and service, Adwords, Google Analytics, Google Translate, a brief discussion on Google Toolkit (including introduction of Google APIs in brief), major features of Google App Engine service.
Use of Amazon Web Services [2L]
Amazon Web Service components and services: Amazon Elastic Cloud, Amazon Simple Storage system, Amazon Elastic Block Store, Amazon SimpleDB and Relational Database Service
Use of Microsoft Cloud Services [2L]
Windows Azure platform: Microsoft's approach, architecture, and main elements, overview of Windows Azure AppFabric, Content Delivery Network, SQL Azure, and Windows Live services

Module 4: Cloud Infrastructure [10L]
Types of services required in implementation – Consulting, Configuration, Customization and Support
Cloud Management [3L]
An overview of the features of network management systems and a brief introduction of related products from large cloud vendors, Monitoring of an entire cloud computing deployment stack – an overview with mention of some products, Lifecycle management of cloud services (six stages of lifecycle)
Live Migration of Virtual Machines: [2L]
Need of Live Migration of Virtual Machine, A Designing Process of Live Migration, and

Security Issues during live migration.
Concepts of Cloud Security [3L]
Infrastructure Security, Infrastructure Security: The Network Level, The Host Level, The Application Level, Data Security and Storage, Aspects of Data Security, Data Security Mitigation Provider Data and Its Security, Identity and Access Management.
Auditing and Compliance in Cloud Environment: [2L]
Data Security in Cloud Computing Environment, Need for Auditing in Cloud Computing Environment, Third Party Service Provider, Cloud Auditing Outsourcing Lifecycle Phases, Auditing Classification.

Module 5: Concepts of Services and Applications [6L]
Service Oriented Architecture: Basic concepts of message-based transactions, Protocol stack for an SOA architecture, Event-driven SOA, Enterprise Service Bus, Service catalogs [6]
Applications in the Cloud: Concepts of cloud transactions, functionality mapping, Application attributes, Cloud service attributes, System abstraction and Cloud Bursting, Applications and Cloud APIs [2]
Cloud-based Storage: Cloud storage definition – Manned and Unmanned. [1]
Webmail Services: Cloud mail services including Google Gmail, Mail2Web, Windows Live Hotmail, Yahoo mail, concepts of Syndication services [1]

Textbooks:

1. Cloud Computing Bible by Barrie Sosinsky, Wiley India Pvt. Ltd, 2013
2. Cloud Computing: A Practical Approach, Anthony T. Velte, Tata Mcgraw-Hill

Reference Books:

1. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, McGraw Hill Education (India)PrivateLimited,2013
2. Fundamentals of Cloud Computing by P. K. Pattnaik, S. Pal, M. R. Kabat, Vikas Publications, 2014.

CO-PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS701D.1	3	3	3	3	2	-	-	-	-	-	-	3	1	1
CS701D.2	3	3	3	3	3	-	-	-	2	-	-	3	1	1
CS701D.3	3	3	3	3	3	-	-	-	1	-	-	3	1	1
CS701D.4	3	2	2	3	3	-	-	-	1	-	-	3	1	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Big Data and Data Analytics
Course Code: CS603C
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites: Data Structure, Design and Analysis of Algorithms, Database Management Systems, Statistics, Artificial Intelligence, Programming skills of Python.

Course Objective(s):

1. Comprehend the fundamental concepts of the Data Analytics exploring machine learning strategies such as Supervised and Unsupervised Learning etc. for analyzing various types of large scale structured as well as unstructured data distributed across multiple locations (Map Reduce, Hadoop and NoSQL Framework).
2. Formulate an engineering problem of analyzing large scale data distributed across multiple locations to make automated meaningful decisions
3. Apply the concepts of Data Analytics to solve problems of making automated decisions dealing with large scale structured as well as unstructured data distributed across multiple locations.
4. Excogitate and Implement ideas to address the challenging issues of Data Analytics.
5. Analyze the effectiveness of various Data Analytics Frameworks.

Course Outcome(s):

After completion of the course students will be able to

CS603C.1: Understand and explain the fundamental concepts of the Big Data Analytics which are primarily explored for making automated decisions using machine learning strategies on analyzing large scale structured as well as unstructured data distributed across multiple locations (Map Reduce, Hadoop and NoSQL Framework) underscoring the utilitarian importance in current technological context for further exploration leading towards lifelong learning.
CS603C.2: Identify and formulate an engineering problem of analyzing large scale data distributed across multiple locations to make automated meaningful decisions within the scope of Big Data Analytics Frameworks.
CS603C.3: Explore relevant literature and apply the concepts of Big Data Analytics to solve problems of making automated decisions dealing with large scale structured as well as unstructured data using Map Reduce, Hadoop and advanced SQL Frameworks.
CS603C.4: Excogitate ideas for proposing solutions to the challenging problems of Big Data Analytics.
CS603C.5: Implement ideas of Big Data Analytics through developing feasible algorithms or frameworks and investigate their effectiveness in solving the relevant problems by analyzing the performances using proper techniques.

Course Content:

Module – 1: Introduction to Basic Analytics [10L]

Introduction: Big data overview, Analyst's perspective on data repositories, Current analytical architecture, Drivers of big data, Examples of big data analytics. Life Cycle of Data Analytics: Phase 1: Discovery, Phase 2: Data preparation, Phase 3: Model planning, Phase 4: Model building, Phase 5: Communication of results, Phase 6: Making operational. Basic Analytic Methods: Visualization, Dirty data, Data exploration versus presentation, Statistical methods for evaluation – hypothesis testing, difference of means, rank sum test, type I and type II errors, ANOVA.

Module - 2: Advanced Analytic Methods I [8L]

Clustering: Overview, K-means, Determining the number of clusters, Diagnostics.
Association Rules: Overview, Apriori algorithm, Evaluation of candidate rules, Application of association rules, Validation and testing, Diagnostics. Regression: Linear regression - model description, Logistic regression – model description, Other regression models. Classification: Decision trees – overview, General algorithm, Decision tree algorithms, Evaluating a decision tree, Naïve Bayes – Bayes theorem, Naïve Bayes classifier, Diagnostics of classifiers.

Module – 3: Advanced Analytic Methods II [8L]

Time Series Analysis: Overview, Box-Jenkins methodology, Autocorrelation function (ACF), Autoregressive model, Moving average model, ARMA and ARIMA model, Building and evaluating an ARIMA model. Text Analysis: Steps in text analysis, Collecting raw text, Representing text, Term Frequency-Inverse Document Frequency (TFIDF), Categorizing documents by types, Determining sentiments. Map Reduce and Hadoop: Analytics for unstructured data – map reduce, Apache Hadoop, Hadoop Ecosystem – Pig, Hive, Hbase, Mahout.

Module – 4: Advanced Analytic Methods III [10L]

Technology and Tools: SQL essentials - Join, Set, Grouping extensions, Advanced SQL – Window functions, User-defined functions, Ordered aggregates, MADlib, NoSQL.
Integration of Techniques: Communicating and operationalizing an analytic project.
Creating final deliverables – Developing core materials, project goals, Main findings, Approach, Model description and model details, Recommendations, Providing technical specifications and code.
Data visualization basics - Key points, evolution of a graph, common representation methods, how to clean up a graphic.

Text book:

1. EMC Education Services (Editor), Data Science and Big Data Analytics. John Wiley & Sons, 2015.
2. Mike Barlow, Real-Time Big Data Analytics: Emerging Architecture. O'Reilly, 2013.

Reference Books:

1. Nathan Marz and James Warren, Big Data: Principles and Best Practices for Scalable Real-time Data Systems. Manning Publications, 2015.
2. Venkat Ankam, Big Data Analytics. Packt Publishing Ltd., UK, 2016.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	1	-	3
CO2	2	3	-	-	-	-	-	-	-	-	-	1	2	3
CO3	2	2	3	2	-	-	-	-	-	-	-	2	2	2
CO4	2	2	2	3	-	-	-	-	-	-	-	3	-	-
CO5	2	2	3	3	2	2	2	-	-	-	-	-	-	-

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Paper Name: Natural Language Processing
Paper Code: CS603D
Contact (Periods/Week): 3L/Week
Credit Point: 3
No. of Lectures: 36

Prerequisite:
1. Solid background in Linear algebra, Probability and Statistics, Artificial Intelligence and Neural Networks.
2. Good Exposure of Python packages.

Course Objective(s)
1. To explore Text Data using various industry standard tools.
2. To explore the Feature Engineering for Text Representation
3. To build the model for Clustering and Classifying Text
4. To implement Machine Learning and Deep Learning techniques.

Course Outcome(s)
On completion of the course, students will be able to

CO1. Understand the basic concepts of text data using various industry standard tools.
CO2. Understand the techniques to do the Feature Engineering for Text Representation.
CO3. Understand the approaches to build models, Clustering and Classifying Text.
CO4. Understand the different techniques in Machine learning and Deep Learning.

Course Content:
Module 1: Natural Language Processing Basics [9L]
What is Natural Language Processing? Different Phases of Natural Language Processing; Linguistics: Language Syntax and Structure, Words, Phrases, Clauses, Grammar, Dependency Grammar, Constituency Grammar, Word-Order Typology; Lemmas and Word forms, Homonyms, Homographs and Homophones, Heteronyms and Heterographs, Polysemes, Capitonyms, Synonyms and Antonyms, Hyponyms and Hypernyms, Stemming and Lemmatization; Representation of Semantics; Text Corpora: Corpora Annotation and Utilities, Accessing Popular Corpora; Parts of Speech Tagging: Training and Building POS Taggers; HMM Part-of-Speech Tagging; NER-Tagging; Relationship Extraction, Temporal Information Extraction, Event Extraction, Template Filling; Conditional Random Fields (CRFs); Shallow Parsing, Chunking; Building Dependency and Constituency Parsers, Application of NLP.

Module 2: Feature Engineering for Text Representation [9L]
Pre-processing the Text Corpus;N-gram Language Models, Smoothing; Traditional Feature Engineering Models; Extracting Features for New Documents; Topic Models in Gensim, LDA, LSI, Hierarchical Dirichlet process; Advanced Feature Engineering Models, Word Embedding, Word2Vec Model, The Continuous Bag of Words (CBOW) Model, The Skip-Gram Model; Semantic Analysis: Exploring WordNet, Understanding Synsets, Analyzing Lexical Semantic Relationships, Semantic Relationships and Similarity, Word Sense Disambiguation.

Module 3: Clustering and Classifying Text [9L]
Clustering text: Text Similarity, Analyzing Term Similarity, Analyzing Document Similarity; Classifying text: Classification Models, Evaluating Classification Models, Building and Evaluating of the Text Classifier; Sentiment Analysis: Text Pre-processing and Normalization, Unsupervised Lexicon-Based Models, Classifying Sentiment with Supervised Learning, Text Summarization, Question & Answering

Module 4: Deep Learning Architectures for Sequence Processing [9L]
Language Models Revisited; Getting words in order with convolutional neural networks (CNNs), Recurrent Neural Networks, Stacked and Bidirectional RNNs; LSTMs and GRUs; Attention, Transformers; Encoder-Decoder Model, Machine Translation; Beam Search; Text Classification using CNNs and LSTM; Chatbots

Text Books:
1. Bhargav Srinivasa-Desikan, “Natural Language Processing and Computational Linguistics", Packt Publishing
2. Dipanjan Sarkar, “Text Analytics with Python", Apress, ISBN-13 (pbk): 978-1-4842-4353-4
3. Daniel Jurafsky, James H. Martin, "Speech and Language Processing", Pearson Education India, Third Edition.
4. Sumit Raj, "Building Chatbots with Python", Apress, ISBN-13 (pbk): 978-1-4842-4095-3

Reference Books:
1. Francois Chollet, “Deep Learning with Python”, Manning Publications; 1st edition
2. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana, "Practical Natural Language Processing", Oreily
3. Hobson Lane, Cole Howard, Hannes Max Hapke, "Natural Language Processing in Action",Manning Publications

CO-PO-PSO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	-	-	1	-	-	1	-	-	1	3	2
CO2	3	3	2	2	-	-	-	-	-	-	-	-	2	3
CO3	3	3	3	3	-	-	2	-	1	-	2	3	3	3
CO4	2	3	-	3	1	-	2	-	-	-	-	3	2	3
CO5	2	3	1	2	-	-	-	2	-	1	1	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Name of the Paper: Mobile Computing
Paper Code: CS604A
Contact (Periods/Week): 3L/Week
Credit Point: 3
No. of Lectures: 36

Prerequisite: Basic understanding of computer networks and digital communication.

Course Objective(s):

- Introduce fundamental concepts and principles of mobile computing.
- Explain wireless communication and networking principles supporting cellular, wireless LANs, and ad hoc networks.
- Understand localization, routing, energy management, and security in mobile systems.

Course Outcome(s):
On completion of the course, students will be able to:

- **CO1:** Analyze wireless communication systems and mobile technologies.
- **CO2:** Apply routing algorithms in infrastructure-based and ad hoc networks.
- **CO3:** Model bandwidth and mobility management in cellular networks.
- **CO4:** Design energy-efficient and secure communication protocols in mobile environments.
- **CO5:** Evaluate recent developments in mobile computing and wireless communication.

Course Content:
Module I: Introduction [6L]
Evolution of mobile communication systems; wireless vs. wired communication. Architecture of cellular networks. Key features of GSM, CDMA, LTE, and 5G. Cellular architecture, Mobility management: handoff types (hard, soft). Bandwidth and energy considerations. Security challenges in mobile environments, Generational evolution from 1G to 5G.

Module II: Wireless Data Communication [8L]
IEEE 802.11: architecture, protocol layers, QoS, roaming support. Bluetooth: piconets, scatternets, protocol stack (L2CAP, RFCOMM). Initialization and self-organization in ad hoc networks. Leader election and routing basics. Energy-efficient MAC protocols. Basic wireless security models, Power consumption in mobile devices. Physical layer: transmission power control. MAC layer: sleep scheduling, contention resolution. Network layer: energy-aware routing. Application layer strategies: data compression and adaptive transmission. Energy-QoS trade-offs.

Module III: Mobility Management [8L]
GSM, Location management techniques: location update, paging. Call establishment in mobile networks. Mobile IP: agent discovery, registration, tunneling. GPRS components and working. Handoff decision strategies. Mobility models: Random Walk, Random Waypoint, Group mobility and their use in simulations.

Module IV: Bandwidth Management [4L]
Frequency spectrum utilization. Channel Assignment Problem (CAP): fixed and dynamic assignment. Graph-based modeling: node and edge coloring techniques. Benchmark instances and theoretical lower bounds. CAP optimization goals and trade-offs.

Module V: Node Localization [4L]
Fundamentals of localization in wireless networks. Outdoor vs. indoor localization challenges. Techniques: Time of Arrival (TOA), Angle of Arrival (AOA), triangulation. Effects of LOS/NLOS signals on accuracy. Beacon-based positioning. Error modeling and mitigation.

Module VI: Ad Hoc Network Communication [6L]
Issues in decentralized communication. MAC protocols for collision avoidance: slotted ALOHA, CSMA/CA. Deterministic scheduling methods. Time slot allocation techniques. Routing protocol classifications: proactive (DSDV), reactive (DSR, AODV), hybrid (ZRP, TORA). Metrics for routing performance.

Text books:

- K. Sinha, S.Ghosh and B. P. Sinha, Wireless Networks and Mobile Computing. CRC Press: New York, 2015.
- J. Schiller, Mobile Communication, Pearson
- Yi-Bing Lin & Imrich Chlamtac, Wireless and Mobile Networks Architectures, John Wiley & Sons, 2001
- Raj Pandya, Mobile and Personal Communication systems and services, Prentice Hall of India, 2001
- XiangYang Li, Wireless Adhoc and Sensor Networks, Cambridge University Press.

Reference books:

- Research articles published on secure wireless communication (authentication, mitigation of DoS, DDoS, eavesdropping) published in leading journals.
- Mark Ciampa, Guide to Designing and Implementing wireless LANs, Thomson learning, Vikas Publishing House, 2001.
- P.Stavronlakis, Third Generation Mobile Telecommunication systems, Springer Publishers.

CO-PO/PSO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	1	1	1	--	--	--	--	3	3	2
CO2	3	3	2	2	1	2	1	--	--	--	--	3	2	2
CO3	3	2	3	3	2	2	1	1	--	--	--	3	3	2
CO4	3	3	3	2	3	2	1	--	--	--	--	3	3	3
CO5	2	3	2	3	1	1	--	1	--	--	--	3	2	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Name of the Paper: Human Computer Interaction
Paper Code: CS604B
Contact (Periods/Week): 3L/Week
Credit Point: 3
No. of Lectures: 36

Prerequisite:
Basic understanding of programming, operating systems, and software engineering concepts.

Course Objective(s):

- Understand the fundamentals of Human-Computer Interaction (HCI).
- Learn usability design and evaluation techniques.
- Explore user-centered design methodologies.
- Analyze the cognitive aspects of interaction.
- Apply principles of interface design for web, mobile, and embedded systems.

Course Outcome(s):

- **CO1:** Explain HCI principles, models, and paradigms.
- **CO2:** Apply cognitive psychology and ergonomics in user interface design.
- **CO3:** Design and prototype usable interfaces using HCI guidelines.
- **CO4:** Evaluate the usability of systems and interfaces.
- **CO5:** Apply interaction techniques in emerging technologies like mobile, VR/AR, and multimodal interfaces.

Course Content:

Module I: Introduction to HCI [4L]
Definition and scope of HCI. Importance of human-centered computing. Historical evolution. Design principles: usability, efficiency, learnability, error tolerance, satisfaction.

Module II: Human Factors and Cognition [5L]
Human perception, memory, attention, learning. Cognitive models (GOMS, KLM). Mental models and metaphors. Ergonomics and limitations in design.

Module III: User-Centered Design (UCD) [5L]
Personas, task analysis, use cases, storyboarding. Participatory design, iterative design, contextual inquiry. Design guidelines and standards.

Module IV: Interaction Styles, Devices, Visual Design and Prototyping [9L]
Command-line, GUI, NUI, touch, gesture, voice-based interactions. Desktop, mobile, wearable, and ubiquitous computing interfaces. Input/output devices and design implications,

Layout, typography, colors, iconography, affordance. Low-fidelity vs high-fidelity prototyping. Tools: Figma, Adobe XD, Balsamiq. Prototyping practices for web and mobile.

Module V: Usability Testing and Evaluation [6L]
Heuristic evaluation, cognitive walkthroughs, usability testing types. A/B testing. Metrics: task time, error rate, satisfaction. Data gathering and analysis.

Module VI: HCI in Emerging Technologies [4L]
VR/AR interfaces, multimodal systems, conversational interfaces. Accessibility and inclusive design. Cross-cultural design considerations.

Module VII: Ethics and Future of HCI [3L]
Ethical implications in design. Privacy and surveillance. Sustainable interaction design. The future of HCI with AI and ambient intelligence.

Text Books

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale – Human-Computer Interaction, Pearson Education.
2. Ben Shneiderman, Catherine Plaisant – Designing the User Interface, Pearson.
3. Don Norman – The Design of Everyday Things, Basic Books.

Reference Books

1. Jenny Preece, Rogers, Sharp – Interaction Design, Wiley.
2. Jeff Johnson – Designing with the Mind in Mind, Morgan Kaufmann.
3. ACM SIGCHI conference proceedings and case studies on usability design.

CO–PO/PSO Mapping

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	2	2	1	-	-	-	-	-	3	2	2
CO 2	3	3	2	2	2	3	1	-	-	-	-	3	3	2
CO 3	3	3	3	3	2	2	1	-	-	-	-	3	3	3
CO 4	2	3	2	3	1	2	2	1	-	-	-	3	3	2
CO 5	3	3	3	2	3	3	1	1	-	-	-	2	2	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: E-Commerce and Digital Business Model
Course Code: CS604C
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Knowledge of Internet Technologies and Web Programming

Course Objective(s):

- To understand the fundamental principles and technologies of e-commerce and digital business models.
- To explore digital business strategy and the role of emerging technologies in reshaping business operations.
- To provide insights into digital payment systems, electronic marketing, and legal & security issues in e-commerce.

Course Outcome(s):

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

CO1: Understand the basic concepts and types of E-Commerce and Digital Business Models.
CO2: Analyze various components and infrastructure required for E-Commerce applications.
CO3: Explore different digital payment systems and security frameworks.
CO4: Evaluate strategies for electronic marketing and customer relationship management.
CO5: Assess legal, ethical, and regulatory issues in E-Commerce.

Course Content:

Module I [6L]
Introduction to E-Commerce:
Definition, History, Benefits and Limitations of E-Commerce, Traditional vs E-Business, E-Commerce Framework, Classification of E-Commerce.

Module II [6L]
Business Models for E-Commerce:
B2B, B2C, C2C, C2B Models, Revenue Models, Value Proposition, Case Studies on successful digital businesses.

Module III [6L]
E-Commerce Infrastructure:

Internet, Intranet, Extranet, EDI, Web Hosting, Cloud Computing, Mobile Commerce, Technology Infrastructure.

Module IV [6L]
Electronic Payment Systems:
Payment Gateways, Digital Wallets, Smart Cards, UPI, Security Issues in Payment Systems, Encryption and SSL, Digital Signatures.

Module V [6L]
E-Marketing and CRM:
Online Marketing Techniques, SEO, SEM, Social Media Marketing, Email Marketing, Personalization, Customer Relationship Management (CRM).

Module VI [6L]
Legal, Ethical and Security Issues:
Cyber Law in India, IT Act 2000, Intellectual Property Rights, Online Fraud and Prevention, Privacy Issues, Security Threats and Countermeasures.

Text book:
1. Kenneth C. Laudon & Carol Guercio Traver, E-Commerce – Business, Technology, Society, Pearson Education.

Reference

1. P. T. Joseph, E-Commerce: An Indian Perspective, PHI Learning.
2. K.K. Bajaj & Debjani Nag, E-Commerce: The Cutting Edge of Business, McGraw Hill.

CO–PO Mapping:

Cos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	2	-	-	-	-	-	-	-	2	1	2
CO2	3	3	2	2	-	-	-	-	-	-	-	2	2	2
CO3	3	3	3	3	2	-	-	-	-	-	-	3	2	2
CO4	2	2	2	3	2	-	-	-	-	-	-	2	3	2
CO5	3	2	2	2	2	2	3	3	-	-	-	2	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Quantum Computing
Course Code: CS604D
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Linear Algebra, Basics of Algorithms, Computational Theory

Course Objective(s):

- To introduce the principles of quantum mechanics relevant to computing.
- To understand quantum computation models and quantum algorithms.
- To explore the practical aspects and challenges of building quantum computers.
- To analyze how quantum computing outperforms classical computing in certain domains.

Course Outcome(s):

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

CO1: Understand the foundational principles of quantum mechanics in the context of computing.

CO2: Analyze quantum gates, circuits, and the model of quantum computation.

CO3: Apply quantum algorithms such as Deutsch-Jozsa, Grover's, and Shor's algorithms.

CO4: Evaluate the advantages and limitations of quantum algorithms over classical ones.

CO5: Discuss the architecture, implementation challenges, and real-world applications of quantum computers.

Course Content:
Module I [6L]
Introduction to Quantum Computing:
Classical vs Quantum computing, Motivation and history, Qubits, Superposition, Quantum entanglement, Postulates of quantum mechanics.

Module II [6L]
Quantum Gates and Circuits:
Single qubit gates (X, Y, Z, H, S, T), Multi-qubit gates (CNOT, Toffoli), Quantum measurement, Circuit representations, Bloch sphere.

Module III [6L]
Quantum Algorithms I:

Deutsch and Deutsch–Jozsa algorithm, Bernstein–Vazirani algorithm, Simon's problem.

Module IV [6L]
Quantum Algorithms II:
Grover's Search algorithm – Quantum advantage in unstructured search, Shor's factoring algorithm – Quantum Fourier Transform (QFT).

Module V [6L]
Quantum Error Correction and Decoherence:
Noise in quantum systems, Quantum error correction codes, Shor code, Bit flip & phase flip codes, Decoherence and fault tolerance.

Module VI [6L]
Quantum Technologies and Applications:
Quantum supremacy, Quantum cryptography (BB84), Quantum teleportation, Quantum hardware: superconducting qubits, trapped ions, and photonic systems.

Text book:
1. Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.

Reference

1. N. David Mermin, Quantum Computer Science: An Introduction, Cambridge University Press.
2. Eleanor Rieffel and Wolfgang Polak, Quantum Computing: A Gentle Introduction, MIT Press.
3. P. Kaye, R. Laflamme, M. Mosca, An Introduction to Quantum Computing, Oxford University Press.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	2	-	-	-	-	-	-	-	2	1	2
CO2	3	3	2	2	-	-	-	-	-	-	-	2	2	2
CO3	3	3	3	3	2	-	-	-	-	-	-	3	2	2
CO4	2	2	2	3	2	-	-	-	-	-	-	2	3	2
CO5	3	2	2	2	2	2	3	3	-	-	-	2	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Cyber Law and Ethics

Course Code: CS605

Contact: 3:0:0

Total Contact Hours: 36

Credits: 3

Prerequisite:

1. Familiarity in computer Networking.
2. Basic concepts about network security.

Course Objectives

- To understand, explore, and acquire a critical understanding of **Cyber Law**
- To learn the fundamentals of **Cybersecurity**
- To develop competencies in identifying and dealing with **frauds and deceptions** (e.g., confidence tricks, scams)

Course Outcomes (COs)

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

- CS605.1: Understand the social and intellectual property issues arising from cyberspace
- CS605.2: Gain knowledge of the Information Technology Act, legal frameworks for privacy, data security, and data protection
- CS605.3: Analyze the interplay between commerce and cyberspace
- CS605.4: Identify and review various network security threats and countermeasures
- CS605.5: Understand and adapt to advanced cybersecurity technologies and issues

Course Contents

Module 1: Introduction to Cybercrime (5 Hours)

- Cybercrime, Forgery, Hacking, Software Piracy
- Computer Network Intrusion, Criminal planning, Passive vs. Active Attacks
- Cyber Stalking

Module 2: Cybercrime in Mobile and Wireless Devices (8 Hours)

- Security challenges in mobile environments
- Cryptographic security for mobile devices
- Attacks on mobile/cell phones: Theft, Viruses, Hacking
- Bluetooth security issues, Laptop malware

Module 3: Tools and Methods Used in Cybercrime (7 Hours)

- Proxy Servers, Password and Random Checking
- Trojan Horses, Backdoors, DoS and DDoS Attacks
- SQL Injection, Buffer Overflow, Script Kiddies, Packaged Defense Tools

Module 4: Cybercrime and Cybersecurity (6 Hours)

- Phishing, Identity Theft, Online Identity Management
- Legal aspects: Indian laws, IT Act, Public Key Infrastructure
- Designing Cybersecurity Policies
- UNCITRAL Model Law, Cyber Jurisdiction (civil, criminal, international)

Module 5: Cyber Ethics (5 Hours)

- Importance of Cyber Law and Cyber Ethics
- Necessity for Cyber Regulations
- Ethics in the Information Society
- Introduction to AI Ethics: Core principles and ethical concerns
- Introduction to Blockchain Ethics

Text Books:

1. ***Cybersecurity*** by Nina Gobole & Sunit Belapune; Pub: Wiley India.
2. **Chris Reed & John Angel**, *Computer Law*, OUP, New York, (2007).
3. **Justice Yatindra Singh**, *Cyber Laws*, Universal Law Publishing Co., New Delhi, (2012).
4. **Verma S.K., Mittal Raman**, *Legal Dimensions of Cyber Space*, Indian Law Institute, New Delhi, (2004).

Reference Books:

1. Kenneth J. Knapp, "Cyber Security and Global Information Assurance: Threat Analysis and Response Solutions", IGI Global, 2009.
2. **Jonathan Rosenoer**, *Cyberlaw: The Law of the Internet*, Springer-Verlag, 1997.
3. **Sudhir Naib**, *The Information Technology Act, 2005: A Handbook*, OUP, New York.
4. **Vasu Deva**, *Cyber Crimes and Law Enforcement*, Commonwealth Publishers, New Delhi, (2003).

CO and PO mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS605.1	1	3	1	1	1	3	-	3	-	-	-	-	1	-
CS605.2	3	3	1	2	3	-	-	3	-	-	-	1	2	-
CS605.3	2	3	3	3	1	1	-	1	-	1	-	2	2	1
CS605.4	2	2	3	3	2	-	-	-	-	-	-	2	3	2
CS605.5	3	2	3	3	3	1	-	2	-	-	-	3	3	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Web and Internet Technology Lab
Course Code: CS691
Contact (Periods/Week): 0:0:3
Credit Point: 1.5

Prerequisites: Computer Networks Lab, C Programming

Course Objective(s):

- To impart the design, development and implementation of Static and Dynamic Web Pages.
- To develop programs for Web using Scripting Languages
- To give an overview of Server-Side Programming in Web.

Course Outcome(s):

After completion of the course students will be able to

CO1: To define interactive web pages using HTML, DHTML, CSS and image map.
CO2: To summarize the knowledge of information interchange formats like XML.
CO3: To implement JavaScript – a client-side scripting languages in web programming.
CO4: To explain PHP applications
CO5: To write the server-side programming concepts using Servlet, JSP.

List of Experiments:

1. Write a single html program through which you can explain a) anchor tag, b)'imp' tag with 'src' attribute, c) paragraph d) heading.
2. Write a single html program through which you can draw a table which consists of 3 row and 4 columns where 1st row contains 4 different column fields of a student's information with red text color and Calibri font style with font 12. Rest cells of whole table contain values with blue text colors and Times new roman font style with font 10.
3. Write a single html program where 1st paragraph can collect its specified style from internal stylesheet describes inside that html program and 2nd paragraph can collect its specified style from another file (external stylesheet).
4. Write a single html program which implements image map concept using 'usemap' and <map>.
5. Write a html program to find out Celsius temperature of a given Fahrenheit temperature using JavaScript.
6. Write a html program to find out m to the power n (m, n valid integer no) using a function using JavaScript.
7. Write a xml parsing technique through which parse a text string into an XML DOM object, and extracts the info from it with JavaScript.

8. Write a simple PHP program through which you can find out maximum and minimum among three nos specified by the user.
9. Write a simple PHP program through which you can implement the concept of GET & POST method w.r.t PHP Form handling.
10. Write a simple program in PHP through which you can create a login page of your own website.
11. Write a simple JSP program through which you can print even and odd no separately within a given range.
12. Create an Online Registration form for individual user of a website using Servlet.

Text book:

1. "Web Technology: A Developer's Perspective", N.P. Gopalan and J. Akilandeswari, PHI Learning, Delhi, 2013. (Topics covered: html, CSS, imagemap, xml)
2. "Learning PHP, MySQL & JavaScript", Robin Nixon, O'Reilly Publication.(Topics covered: PHP, Java Script)
3. "Head First Servlet's & JSP", Bryan Basham, Kathy Sterra, Bert Bates, O'Reilly Publication. (Topics covered: Servlet, JSP)

Reference

1. "Programming the World Wide Web", Robert. W. Sebesta, Fourth Edition, Pearson Education.
2. "Core Web Programming"- Second Edition-Volume I and II, Marty Hall and Larry Brown, Pearson Education,

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	-	-	-	-	1	3	2	1	2
CO2	2	2	3	2	3	-	-	-	-	2	2	2	2	3
CO3	1	2	1	2	2	-	-	-	-	-	2	2	3	-
CO4	3	2	3	2	3	-	-	-	-	1	2	3	-	1
CO5	1	3	2	3	2	-	-	-	-	-	1	2	2	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Deep Learning Lab
Course Code: CS692
Contact (Periods/Week): 0:0:3
Credit Point: 3
No. of Lectures: 36

Prerequisite: A solid background in Statistics, Calculus, Linear Algebra, and Probability. Good exposure to Python packages like NumPy, Pandas, Matplotlib, and Scikit-learn.

Course Objectives:

- To introduce the Keras and TensorFlow APIs for implementing Neural Networks.
- To build and train Convolutional Neural Network (CNN) models for computer vision tasks.
- To build and train Recurrent Neural Network (RNN) models for sequence analysis.
- To implement advanced architectures, including generative models and Transformers.

Course Outcome(s):
After completion of the course students will be able to

CO1: Understand and utilize the Keras/ TensorFlow API in Python for building deep learning models.
CO2: Implement, train, and evaluate Convolution Neural Network models for image-based applications.
CO3: Implement, train, and evaluate Recurrent Neural Network models for handling sequential data.
CO4: Design and implement generative models, such as Auto-encoders and Generative Adversarial Networks (GANs).

CO5: Apply advanced techniques like transfer learning, functional APIs, and Transformer models to solve complex problems.

List of Assignments (12 Experiments)

1. Lab Environment Setup and "Hello, World!" of Deep Learning
   - Task: Configure a Python environment with TensorFlow and Keras. Build and train a simple neural network on a dataset like MNIST.
2. Implementing a Deep Neural Network for a Regression Task
   - Task: Build, train, and evaluate a fully-connected deep neural network to predict a continuous value (e.g., house price prediction).
3. Implementing a Deep Neural Network for Multi-Class Classification
   - Task: Develop a deep neural network to classify data into multiple categories (e.g., Iris flower classification).
4. Building Convolution Neural Networks (CNNs) for Image Recognition
   - Task: Construct a CNN from scratch to classify images from a standard dataset like CIFAR-10.

5. Applying Transfer Learning with Pre-trained CNN Models
   - Task: Use a pre-trained model (e.g., VGG16, ResNet50) and fine-tune it for a custom image classification task to improve performance.
6. Hyperparameter Tuning and Model Optimization
   - Task: Experiment with different optimizers, learning rates, batch sizes, and use techniques like Keras Tuner to find the best hyperparameters for a model.
7. Working with Sequence Data using Recurrent Neural Networks (RNNs)
   - Task: Implement a simple RNN to perform a sentiment analysis task on text data.
8. Advanced Sequence Modeling: Text Generation using LSTMs/GRUs
   - Task: Build and train a character-level LSTM or GRU model to generate new text based on a sample corpus.
9. Implementing Autoencoders for Denoising and Feature Extraction
   - Task: Construct a deep autoencoder to remove noise from images and visualize the compressed latent space representations.
10. Building a Generative Adversarial Network (GAN) for Image Generation
    - Task: Implement a simple GAN to generate new, synthetic images (e.g., handwritten digits).
11. Advanced Model Architectures with the Keras Functional API
    - Task: Build a model with multiple inputs and outputs using the Keras Functional API to solve a more complex problem.
12. Introduction to Transformer Models for a NLP Task
    - Task: Implement a basic Transformer model or use a pre-trained one (like BERT from Hugging Face) for a text classification or question-answering task.

Text Books

1. Jojo Moolayil, "Learn Keras for Deep Neural Networks", Apress, ISBN-13 (pbk): 978-1-4842-4239-1
2. Francois Chollet, "Deep Learning with Python", Manning Publications; 2nd edition

CO–POMapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	3	3	3	-	-	-	-	-	1	-	-	-
CO2	3	2	2	2	3	-	-	-	-	-	-	-	-	-
CO3	3	2	3	3	3	-	-	-	-	2	-	-	-	-
CO4	3	3	3	2	3	-	-	-	-	-	1	-	-	-
CO5	3	3	3	3	3	-	-	-	1	2	2	-	-	-

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Image Processing Lab
Course Code: CS693A
Contact (Periods/Week): 0:0:3
Credit Point: 1.5

Prerequisite: Should have prior knowledge on syntaxes of programming like PYTHON, JAVA.

Course Objective(s)

- To learn discrete Fourier transform and its properties
- To study the monochrome and color image fundamentals
- To learn the analytical tools and methods which are currently used in digital image processing as applied to image information for human viewing.
- To learn image compression and segmentation techniques.

Course Outcomes:
On completion of the course students will be able to

CO1: Acquire the fundamental concepts of a digital image processing system such as image acquisition, enhancement, segmentation, transforms, compression, morphology, representation and description.
CO2: Analyze images in the spatial domain.
CO3: Analyze images in the frequency domain through the Fourier transform.
CO4: Design and implement with MATLAB algorithms for digital image processing operations such as point processing, histogram processing.
CO5: Spatial and frequency domain filtering, denoising, transforms, compression, and morphological processing.

Experiments:

1. W.A.P in MATLAB to extract different attributes of an Image.
2. W.A.P in MATLAB program for Image Negation.
3. W.A.P in MATLAB for Power Law Transformation.
4. W.A.P in MATLAB for Histogram Mapping and Equalization.
5. W.A.P in MATLAB for Image Smoothening and Sharpening.
6. W.A.P in MATLAB for Edge Detection using Sobel, Prewitt and Roberts Operators.
7. W.A.P in MATLAB for Morphological Operations on Binary Images.
8. W.A.P in MATLAB for Pseudo Colouring of images.
9. W.A.P in MATLAB for Chain Coding applied on images.
10. W.A.P in MATLAB for DCT/IDCT Computation.

Text Books:

- Practical Machine Learning and Image Processing Himanshu Singh, Apress,2018
- Digital Image Processing: Practical Implementation with MATLAB, by Dr. A Chrispin Jiji, Dr. Y R Annie Bessant, et al. | 29 December 2023

Reference books:

- **Digital Image Processing, 4e Paperback – 30 July 2018,by Rafael C. Gonzalez (Author), Richard E. Woods (Author), Pearson**

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	-	-	2	-	1	3	-	-	-	-	3	3	3
CO2	-	1	3	1	-	-	-	-	-	1	-	3	3	3
CO3	1	2	-	-	-	3	-	-	1	-	2	3	3	3
CO4	2	-	2	-	-	1	-	-	-	1	1	3	3	3
CO5	-	2	-	3	-	1	-	1	-	1	-	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Cloud Computing Lab
Course Code: CS693B
Contact (Periods/Week): 0:0:3
Credit Point: 1.5

Prerequisite:

- Should have the basic knowledge of Operating Systems.
- Should be aware of the fundamental concepts of Networking.
- Should have knowledge of heterogeneous systems and resource management.

Course Objective(s):

- To impart hands-on experience on deploying and managing virtual machines and services using popular cloud platforms (AWS, Azure, GCP). (Apply – Level 3)
- To develop practical skills in implementing virtualization, resource provisioning, and load balancing in cloud environments. (Apply – Level 3)
- To enable students to analyze and evaluate different service models (IaaS, PaaS, SaaS) and deployment strategies through lab experiments. (Analyze/Evaluate – Level 4/5)
- To design and develop secure, scalable, and efficient cloud-based applications addressing real-world challenges. (Create – Level 6)
- To enhance technical documentation, teamwork, and communication skills by preparing reports and presentations on lab outcomes. (Apply/Communicate – Level 3)

Course Outcomes:
On completion of the course students will be able to

CO	Course Outcome
CS693B.1	Demonstrate the use of various cloud platforms (e.g., AWS, Azure, GCP) to deploy and manage virtual instances and services. (Apply – Level 3)
CS693B.2	Implement virtualization techniques and cloud resource management methods including load balancing and scheduling. (Apply – Level 3)
CS693B.3	Analyze and compare the features of IaaS, PaaS, and SaaS through hands-on experimentation. (Analyze – Level 4)
CS693B.4	Develop secure and scalable cloud-based applications with attention to privacy, security, and performance. (Create – Level 6)
CS693B.5	Evaluate cloud-based solutions and prepare comprehensive technical reports reflecting industry practices. (Evaluate – Level 5)

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Content:

Module 1 Introduction to Cloud Environment and Virtualization: Installing and configuring virtual machine tools such as VirtualBox or VMware, creating and managing virtual machines, understanding virtualization concepts including host and guest systems, managing VM disk, memory, and network settings, exploring cloud simulation tools like CloudSim or OpenStack for lab environment setup.

Module 2 Understanding Cloud Service Models: Hands-on implementation of IaaS using AWS EC2 or Azure Virtual Machines, exploring PaaS by deploying applications using Google App Engine or Heroku, identifying SaaS platforms and performing basic operations using services like Google Workspace or Microsoft 365, distinguishing between the deployment and functional aspects of each service model.

Module 3 Working with Cloud Storage and Resource Allocation: Creating cloud storage buckets and objects using platforms like AWS S3 or Azure Blob Storage, managing permissions and storage classes, using CLI or SDKs to interact with cloud storage, allocating and managing computing resources, understanding elasticity and auto-scaling.

Module 4 Virtualization and Load Balancing Techniques: Exploring different types of virtualization such as full, para, and hardware-assisted, creating and managing containers using Docker, implementing basic load balancing using AWS ELB or NGINX, distributing traffic among multiple instances, testing application performance under load-balanced conditions.

Module 5 Identity Management and Cloud Security: Implementing identity and access management using IAM services, defining and assigning roles and permissions to users and groups, configuring firewall and security group rules, encrypting data at rest and in transit, setting up multi-factor authentication and performing basic auditing tasks.

Module 6 Development and Deployment of Cloud-Based Applications: Designing and deploying a full-stack or modular cloud application using PaaS platforms, integrating storage, compute, and database services, documenting the deployment pipeline, generating comprehensive lab reports, presenting the final application along with challenges and solutions through viva voce and demonstrations.

Textbooks:

- Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, McGraw Hill Education (India)PrivateLimited,2013

- Fundamentals of Cloud Computing by P. K. Pattnaik, S. Pal, M. R. Kabat, Vikas Publications, 2014.

Reference Books:

- Cloud Computing Bible by Barrie Sosinsky, Wiley India Pvt. Ltd, 2013
- Cloud Computing: A Practical Approach, Anthony T. Velte, Tata Mcgraw-Hill

CO-PO-PSO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS693B.1	3	3	2	3	3	-	-	-	1	2	1	3	2	3
CS693B.2	3	3	3	3	2	-	-	-	-	1	-	3	2	3
CS693B.3	3	3	3	3	2	-	-	-	1	1	-	2	3	2
CS693B.4	3	3	3	3	2	2	2	1	2	2	2	2	3	3
CS693B.5	3	2	2	2	3	-	-	-	2	3	2	2	2	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Data Analytics Lab
Course Code: CS693C
Contact (Periods/Week): 0:0:3
Credit Point: 1.5

Prerequisite:

1. **Knowledge of Data Structures and Algorithms** – for understanding data organization and complexity analysis.
2. **Understanding of Database Management Systems (DBMS)** – to handle structured data and query optimization.
3. **Basic Concepts in Statistics and Probability** – essential for data interpretation, hypothesis testing, and model evaluation.
4. **Familiarity with Machine Learning Fundamentals** – including supervised and unsupervised learning approaches.
5. **Programming Proficiency in Python** – particularly in libraries like NumPy, Pandas, Scikit-learn, and Matplotlib.
6. **Exposure to Big Data Tools** – such as Hadoop, MapReduce, and NoSQL databases, for distributed data handling.

Course Objective(s):

- To provide hands-on experience in implementing data analytics techniques for processing structured and unstructured data. (Apply – Level 3)
- To develop practical skills in using machine learning algorithms such as clustering, classification, and regression. (Apply – Level 3)
- To train students in applying statistical methods and data visualization tools for analytical interpretation. (Analyze – Level 4)
- To familiarize students with big data tools like Hadoop, MapReduce, Pig, Hive, and NoSQL for large-scale data processing. (Apply – Level 3)
- To guide students in developing end-to-end analytics projects including data preparation, model building, evaluation, and result presentation. (Create – Level 6)

Course Outcomes:
On completion of the course students will be able to

CO	Course Outcome
CS693C.1:	Implement basic data analytics operations including data cleaning, transformation, and exploratory analysis using Python and statistical tools. (Apply – Level 3)
CS693C.2:	Apply machine learning techniques such as clustering (K-Means), classification (Decision Trees, Naïve Bayes), and regression models on real-world datasets. (Apply – Level 3)

CS693C.3: Analyze time-series data and textual data using relevant libraries and frameworks such as ARIMA, TF-IDF, and sentiment analysis. (Analyze – Level 4)

CS693C.4: Use Hadoop and its ecosystem (Pig, Hive, HBase) to perform distributed data storage, querying, and processing tasks. (Apply – Level 3)

CS693C.5: Develop and present comprehensive data analytics projects, including result interpretation through effective visualization and reporting techniques. (Create – Level 6)

Course Content:

Module 1 Data Pre-processing and Exploratory Data Analysis (EDA): Loading and handling datasets using Python libraries (NumPy, Pandas), handling missing values, outlier detection, data transformation, feature scaling, encoding categorical variables, basic data visualization using Matplotlib and Seaborn.

Module 2 Supervised Learning Techniques: Implementing classification algorithms such as Decision Trees, Logistic Regression, and Naïve Bayes; regression models like Linear Regression; evaluating model performance using accuracy, confusion matrix, precision, recall, and F1-score on real-world datasets.

Module 3 Unsupervised Learning Techniques: Performing clustering using K-Means, Hierarchical Clustering, and DBSCAN; understanding the elbow method and silhouette score for optimal cluster selection; visualizing clustering outputs.

Module 4 Time Series and Text Data Analysis: Performing time-series forecasting using ARIMA and related models; plotting autocorrelation and partial autocorrelation; analyzing text data using TF-IDF and sentiment analysis techniques; classifying text using Naïve Bayes.

Module 5 Big Data Processing Using Hadoop Ecosystem: Executing basic MapReduce programs; using Pig Latin for data querying; implementing queries in Hive; working with NoSQL databases such as HBase or MongoDB to perform basic CRUD operations on large datasets.

Module 6 Capstone Project and Report Preparation: Developing a mini-project based on any real-world dataset integrating multiple analytics techniques; documenting the problem statement, methodology, implementation, results, and conclusion; preparing a technical report and presenting the project outcomes through viva and demonstration.

Text book:

1. Nathan Marz and James Warren, Big Data: Principles and Best Practices for Scalable Real-time Data Systems. Manning Publications, 2015.

2. Venkat Ankam, Big Data Analytics. Packt Publishing Ltd., UK, 2016.

Reference Books:

1. EMC Education Services (Editor), Data Science and Big Data Analytics. John Wiley & Sons, 2015.
2. Mike Barlow, Real-Time Big Data Analytics: Emerging Architecture. O'Reilly, 2013.

CO-PO-PSO Mapping:

Cos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CS693C.1	3	3	2	2	2	-	-	-	1	2	1	2	2	2
CS693C.2	3	3	3	3	2	-	-	-	1	1	-	3	2	3
CS693C.3	2	3	3	3	3	-	-	-	1	2	-	2	3	3
CS693C.4	2	2	3	3	2	2	-	-	1	2	2	2	2	3
CS693C.5	2	2	3	3	3	2	2	1	2	3	2	3	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Natural Language Processing Lab
Course Code: CS693D
Contact (Periods/Week): 0:0:3
Credit Point: 1.5

Prerequisite: Familiarity with the programming concepts in any language. A solid background in mathematics, including probability and statistics.

Course Objective(s):

- To learn the basics of NLTK toolkit
- To learn the principles of NLP through programming
- To build an application using different algorithms and natural language processing techniques.

Course Outcome(s):
CO1: To identify text corpora, lexical resources and process of raw text.
CO2: To write structured programs for categorizing and tagging of words, segmentation of sentences.
CO3: To implement different morphological methods on text and extract information from it.
CO4: To express sentence structure, build feature-based grammar, meaning of sentences and to manage linguistic data.

Course Content:
Module I:
Introduction to list, dictionaries etc., input and output handling, saving data to files, retrieving data from files, Writing functions and code reusing.

Module II:
Introduction to working knowledge of matplotlib, Sci-Kit, NumPy and other necessary tools andlibraries as per the need, Language processing with python, Manipulating texts and words by writing programs.

Module III:
Accessing text corpora, lexical resources, using WordNet through NLTK tool kit, Processing raw text, normalizing, segmenting, applying regular expressions, Writing programs to categorize texts, words, tagging words using tagger, generating tagged tokens, NGram tagging, text classification.

Module IV:
Writing programs to extract information from texts, Writing programs to analyze sentence, its meaning etc., Managing linguistic data through programs.

Textbooks:
1. Steven Bird, Ewan Klein, and Edward Loper. "Natural Language Processing– Analyzing Text with theNatural Language Toolkit". 2009, O'Reilly, 1ed.

Reference books:
1. Learning Python: Powerful Object-Oriented Programming: 5th Edition by Mark Lutz, 2013, O'Reilly.
2. Natural Language Toolkit documentation (https://www.nltk.org/)

CO-PO/PSO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	2	2	-	1	-	-	-	2	-	1	-	1
CO2	1	2	2	2	-	-	2	3	3	2	1	-	1	2
CO3	-	2	3	3	3	1	1	-	2	3	-	2	1	-
CO4	2	2	2	2	3	1	2	-	-	1	3	2	1	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

4th Year 7th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
A. THEORY									
1	ENGG	Major	CS701A	Blockchain Technology	3	0	0	3	3
			CS701B	Optimization Technique					
			CS701	Bio-informatics					
			CS701D	Robotics					
2	HUM	Minor	HU(CS)701	Human Resource Development and Organizational Behavior	2	0	0	2	2
3	HUM	Value Added Course	HU702	Research Methodology & IPR	1	0	0	1	1
B. PRACTICAL									
1	PRJ	Project	CS793	Project-III	0	0	12	12	6
2	ENGG	Internship	CS781	Internship (Minimum 1 Month)	0	0	0	0	2
3	ENGG	Skill Enhancement Course	PR792	Rapid Prototyping Lab	0	0	0	4	2
Total of Theory, Practical								**22**	**16**

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Blockchain Technology
Course Code: CS701A
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Distributed DBMS, Cryptography, C/Python Programming

Course Objective(s):

- To assess blockchain applications in a structured manner.
- To impart knowledge in block chain techniques and able to present the concepts clearly and structured.
- To get familiarity with future currencies and to create own crypto token.

Course Outcome(s):
After completion of the course students will be able to

CO1: To explain the concepts of Blockchain in distributed environment by introducing Smart Contracts.
CO2: To apply different consensus mechanism in Blockchain by discussing detailed architecture.
CO3: To analyze proof of work and proof of stake w.r.t bitcoin transaction and scripting.
CO4: To develop the applications of Smart Contacts in Ethereum Blockchain.
CO5: To evaluate various privacy and Security issues along with different attacks in Blockchain.

Course Content:

Module 1: Introduction to Blockchain [6L]
History of Blockchain; Concepts of Blocks, ledger in Blockchain; Characteristics of Blockchain[2]; Difference between centralized and de-centralized system; concept of Distributed Ledger Technology and Hashing in blockchain[2]; Smart Contract; Different types of Blockchain- Public, Private, Consortium and Hybrid; Importance and applications of Blockchain in various domains[2].

Module -2: Blockchain Architecture [10L]
Structure of Blockchain; Working procedure of blockchain [2]; Consensus mechanism - Proof of Work (PoW), Proof of Stake (PoS), Deligated Proof of Stake (PoS); Proof of Authority (PoA), Round Robin and Proof of Elapsed Time (PoET) [4]; Byzantine Fault Tolerance (BFT),Merkle Tree and Merkle Root; Consensus Model, Incentive Model[4].

Module 3: Bitcoin and Scripting [8L]
Concept of Transactions, Wallets and Mining; Applications of Bitcoin [1]; BitCoin-Introduction to Bitcoin; Bitcoin wallets and addresses; Block Structure and Transactions

Structure of BitCoin[2]; Transactions types; Bitcoin network[1]; proof of work and Proof of stake in Bitcoin[2]; Bitcoin consensus, Bitcoin scripting language[2].

Module 4: Ethereum and Smart Contacts [7L]
Overview, Accounts, Transactions, Concept of Smart Contracts[2]; Life Cycle of Smart Contracts; Benefits of smart contracts; Difference between Traditional and Smart Contracts[2]; Structure of Ethereum; Virtual Machine (EVM); Gas Mechanism; Case study of Supply Chain in Ethereum[3].

Module 5: Privacy and Security issues in Blockchain [5L]
Various privacy and security issues in Blockchain[1]; Pseudo-anonymity vs. anonymity; Zcash and ZkSNARKS for anonymity preservation[2]; Attacks on Blockchains - Sybil attacks, 51% attacks, Double Spending Attack, Eclipse Attack and Smart Contract Exploits[2].

Text book:
1. Blockchain Technology: Algorithms and Applications by Asharaf S,Sivadas Neelima, et al, Willey Publications, 2023
2. Imran Bashir, Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition, Birmingham: Packt Publishing
3. Arvind Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press, 2016

Reference Book
1. Imran Bashir**,** "Mastering Blockchain: Inner workings of blockchain, from cryptography and decentralized identities, to DeFi, NFTs and Web3", 4th Edition-Packt Publishing, 2023
2. Josh Thompson, 'Blockchain: The Blockchain for Beginnings, Guild to Blockchain Technology and Blockchain Programming', Create Space Independent Publishing Platform, 2017.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	2	3	2	-	-	-	-	-	2	3	1	2
CO2	2	3	3	1	3	-	-	-	-	-	3	2	2	2
CO3	3	2	2	2	2	-	-	-	-	-	1	2	3	-
CO4	3	2	2	2	3	-	-	-	-	-	2	3	2	1
CO5	2	3	2	3	1	-	-	-	-	-	1	2	2	2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: OPTIMIZATION TECHNIQUES
Course Code: CS701B
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Basic Knowledge of Function, plotting of Equation and in equations, Formulation of Mathematical Problem. Finding maximum and minimum from row or column or from Matrix.

Course Objective(s):
Purpose of this course to develop models and then analyze the model using the techniques of Operations Research, Decision making under uncertainty and risk.

Course Outcome(s):
After completion of the course students will be able to
CO1: Recall the distinctive characteristics of different types of decision-making problem to formulate and solve a real-world problem a prototype of mathematical problem.
CO2: Understand the theoretical workings of appropriate decision-making approaches and tools to identify the optimal strategy in competitive world
CO3: Apply the principles of different Methods/Model of Operations Research to solve practical problems.
CO4: Analyze different engineering problems linked with Optimization Technique.

Course Content:
Module I [10L]
Linear Programming Problem(LPP): Basics of Linear Programming Problem(LPP) and its Applications. General Mathematical Formulation of LPP; Definitions: Convex set, Solution, Feasible Solution, Basic and Non-Basic Variables, Basic Feasible Solution, Degenerate and Non-Degenerate solution, Optimum/Optimal Solution; Solution of LPP by Graphical Analysis/Method, Simplex Method, Charnes' Big M-Method; Duality Theory.

Module II [6L]
Transportation Problem, Assignment Problem

Module III [5L]
Game Theory: Introduction; Two person Zero Sum game, Saddle Point; Mini-Max and MaxiMin Theorems (statement only) and problems; Games without Saddle Point; Graphical Method; Principle of Dominance.

Module IV [5L]
Network Optimisation Models: CPM PERT (Arrow network), Time estimates, earliest expected time, latest allowable occurrence time, latest allowable occurrence time and stack. Critical path, Probability of meeting scheduled date of completion of project. Calculation of

CPM network. Various floats for activities.

Module V [2L]
Sequencing: Johnson's Algorithm (1957) For n Jobs and two machines, n Jobs and three machines.

Module VI [5L]
Queuing Theory: Introduction and Basic Structure of Queuing Theory; Basic Definations and Notations; Birth-and-Death Model (Poisson / Exponential distribution); Poisson Queue Models: (M/M/1): (∞/FIFO)and(M/M/1):(N/FIFO) and Problems.

Module VII [3L]
Inventory Control: Determination of EOQ, Components, Deterministic Continuous & Deterministic Periodic Review Models, Stochastic Continuous & Stochastic Periodic Review Models.

Text book:
1. An introduction to Optimization by Edwin P K Chong, Stainslaw Zak
2. Nonlinear Programming by Dimitri Bertsekas
3. Operations Research by Kanti Swaroop and P.K. Man Mohan, Sultan Chand and Sons
4. Linear Programming and Theory of Games by Ghosh and Chakraborty, Central Book Agency
5. Linear Programming and Theory of Games by P.M.Karak, ABS Publishing House
6. Operations Research, D.K.Jana & T.K.Roy, Chhaya Prakashani Pvt. Ltd.
7. Operations Research, Kalavati,VIKAS 6. Operations Research,Humdy A Taha,PHI / Pearson.

Reference
1. Operations Research Theory and Applications by J.K.Sharma, Macmillan India Limited.
2. Operations Research, Vijayakumar, Scitech
3. Operations Research by S.D. Sharma, Kedar Nath Ram Nath Publishers.
4. Operations Research by A.P. Verma, S. K. Kataria & Sons.
5. Operations Research by P.K. Gupta & Hira, S.Chand

CO–PO-PSO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	-	1	-
CO2	3	2	3	-	-	-	-	-	-	-	-	1	2	-
CO3	3	2	3	2	-	-	-	-	-	-	-	2	-	-
CO4	2	3	3	1	-	-	-	-	-	-	-	-	-	-

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Bio-informatics
Course Code: CS701
Contact (Periods/Week):3:0:0
Credit Point: 3
No. of Lectures: 36

Prerequisites: Mathematics & Statistics, Database, Biology at 12 levels.

Course Objective(s):

- Be familiar with the modelling techniques.
- Learn microarray analysis.
- Exposed to Pattern Matching and Visualization.

Course Outcome(s):

CO1 To acquire the knowledge of Bioinformatics technologies with the related concept of DNA, RNA and their implications
CO2 To develop idea in MOLECULAR BIOLOGY
CO3 To understand the concept and techniques of different types of Data Organization and Sequence Databases with different types of Analysis Tools for Sequence Data Banks
CO4 To acquire the knowledge of the DNA SEQUENCE ANALYSIS
CO5 To analyze the performance of different types of Probabilistic models used in Computational Biology

Course Content:

Module -1: [7L]

INTRODUCTION TO MOLECULAR BIOLOGY:
Concepts of Cell, tissue, types of cell, components of cell, organelle. Functions of different organelles. Concepts of DNA: Basic Structure of DNA; Double Helix structure; Watson and crick model. Exons and Introns and Gene Concept.
Concepts of RNA: Basic structure, Difference between RNA and DNA. Types of RNA.
Concept of Protein: Basic components and structure. Introduction to Central Dogma: Transcription and Translation, Introduction to Metabolic Pathways.
Introduction to Bioinformatics. Recent challenges in Bioinformatics.

Module -2: [10L]

Introduction to Genomic data, Data Organization and Sequence Databases: Sequence Data Banks - Introduction to sequence data banks - protein sequence data bank. Signal peptide data bank, Nucleic acid sequence data bank - GenBank, AIDS virus sequence data bank. RRNA data bank, structural data banks - protein Data Bank (PDB), The Cambridge Structural Database (CSD) : Genome data bank - Metabolic pathway data: Microbial and Cellular Data Banks.
Introduction to MSDN (Microbial Strain Data Network):Numerical Coding Systems of Microbes, Hibridoma Data Bank Structure, Virus Information System Cell line information system; Protein Sequence Databases, DNA sequence databases. sequence database search

programs like BLAST and FASTA. NCBI different modules: GenBank; OMIM, Taxonomy browser, PubMed;

Module 3: [8L]
DNA SEQUENCE ANALYSIS
DNA Mapping and Assembly: Size of Human DNA, Copying DNA: Polymerase Chain Reaction (PCR), Hybridization and Microarrays, Cutting DNA into Fragments, Sequencing Secondary Structure predictions; prediction algorithms; Chao-Fasman algorithm, Hidden-Markov model, Neural Networking. Tertiary Structure predictions; prediction algorithms; Chao-Fasman algorithm, Hidden-Markov model, Neural Networking.

Module -4: [10L]
Introduction Probabilistic models used in Computational Biology: Probabilistic Models; Gene Regulatory Method Application of HMM in Bioinformatics: Gene finding, profile searches, multiple sequence alignment and regulatory site identification. Applications in Biotechnology: Protein classifications, Fold libraries, Protein structure prediction: Fold recognition (threading), Protein structure predictions: Comparative modelling (Homology), Advanced topics: Protein folding, Protein-ligand interactions, Molecular Modelling & Dynamics, Drug Designing.

Text book:
1. Yi-Ping Phoebe Chen (Ed), “Bio Informatics Technologies”, First Indian Reprint, Springer Verlag, 2007.
2. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins" by Andreas D. Baxevanis & B.F. Francis Ouellette.

Reference
1. Bryan Bergeron, “Bio Informatics Computing”, Second Edition, Pearson Education, 2003.
2. Arthur M Lesk, “Introduction to Bioinformatics”, Second Edition, Oxford University Press, 2005

CO–PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3					1	1				
CO2		1	2	1						1	
CO3	1	2		2	2				1		
CO4	2					2	2			1	1
CO5		3		1		3		1			2

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Robotics
Course Code: CS701D
Contact (Periods/Week): 3:0:0
Credit Point: 3
No. of Lectures: 36L

Prerequisite: Microprocessor & Microcontroller, Computer Organization &Architecture

Course Objective(s):

- To study microcontroller operations for robotics
- To study how different interfaces are implemented in a microcontroller.
- To learn how Microchip PIC micro PIC16F627 can be erased and reprogrammed
- To learn how different sensors, outputs, and peripherals can be wired to a microcontroller to work cooperatively and create a high-level control program.
- To design robots in a real time environment.

Course Outcome(s):

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

CO1 To describe and explain the microcontrollers used the in robots.
CO2. To design the software and build the prototype of robots.
CO3. To apply localization and mapping aspects of mobile robotics.
CO4. To demonstrate self-learning capability.

Course content:

Module 1: [5L]

Introduction -- brief history, types, classification and usage, Science and Technology of robots, Some useful websites, textbooks and research journals.

Position and orientation of a rigid body, Homogeneous transformations, Representation of joints, link representation using D-H parameters, Examples of D-H parameters and link transforms, different kinds of actuators – stepper, DC servo and brushless motors, model of a DC servo motor, Types of transmissions, Purpose of sensors, internal and external sensors, common sensors–encoders, tachometers, strain gauge based force-torque sensors, proximity and distance measuring sensors, and vision.

Module 2: [8L]

Kinematics of serial robots: - Direct and inverse kinematics problems, Examples of kinematics of common serial manipulators, workspace of a serial robot, Inverse kinematics of constrained and redundant robots, Tractrix based approach for fixed and free robots and multi-body systems, simulations and experiments, Solution procedures using theory of elimination, Inverse kinematics solution for the general 6R serial manipulator.

Kinematics of parallel robots: - Degrees-of-freedom of parallel mechanisms and manipulators, Active and passive joints, Constraint and loop-closure equations, Direct kinematics problem, Mobility of parallel manipulators, Closed-from and numerical solution, Inverse kinematics of parallel manipulators and mechanisms, Direct kinematics of Gough-

Stewart platform.

Module 3: [8L]
Velocity and static analysis of robot manipulators:- Linear and angular velocity of links, Velocity propagation, Manipulator Jacobians for serial and parallel manipulators, Velocity ellipse and ellipsoids, Singularity analysis for serial and parallel manipulators, Loss and gain of degree of freedom, Statics of serial and parallel manipulators, Statics and force transformation matrix of a Gough-Stewart platform, Singularity analysis and statics.
Dynamics of serial and parallel manipulators - Mass and inertia of links, Lagrangian formulation for equations of motion for serial and parallel manipulators.

Module 4: [9L]
Motion planning and control - Joint and Cartesian space trajectory planning and generation, Classical control concepts using the example of control of a single link, Independent joint PID control, Control of a multi-link manipulator, Non-linear model based control schemes, Simulation and experimental case studies on serial and parallel manipulators, Control of constrained manipulators, Cartesian control, Force control and hybrid position/force control, Advanced topics in non-linear control of manipulators.

Module 5: [6L]
Modeling and analysis of wheeled mobile robots - Introduction and some well-known wheeled mobile robots (WMR), two and three-wheeled WMR on flat surfaces, Slip and its modelling, WMR on uneven terrain, Design of slip-free motion on uneven terrain, Kinematics, dynamics, and static stability of a three-wheeled WMR ‘s on uneven terrain, Simulations in MATLAB and ADAMS.
Selected advanced topics in robotics:- Introduction to chaos, Non-linear dynamics and chaos in robot equations, Simulations of planar 2 DOF manipulators, Analytical criterion for unforced motion. Gough-Stewart platform and its singularities, use of near singularity for fine motion for sensing, design of Gough-Stewart platform based sensors. Over-constrained mechanisms and deployable structures, Algorithm to obtain redundant links and joints, Kinematics and statics of deployable structures with pantographs or scissor-like elements (SLE‘s).

Textbooks:
MykePredko—ProgrammingRobotControllers–McGrawHill,1st edition,2003.

Reference books:
1. Michael slater, Microprocessorbased design: Acomprehensive Guideto Effective Hardware Design, Prentice Hall, 1989.
2. MykePredko, Programmingandcustomizingthe8051-micro-controller, Tata McGraw-Hill, New Delhi, 2000.

CO-PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C605C.1	3	3	-	1	1	-	-	-	-	-	-	-	1	1
C605C.2	2	3	-	1	-	-	-	-	-	-	-	-	1	1
C605C.3	2	3	3	-	-	-	-	-	-	-	-	-	1	1
C605C.4	2	2	-	-	-	-	-	-	-	-	-	3	2	1

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Human Resource Development and Organizational Behavior
Course Code: HU(CS)701A
Contact (Periods/Week): 2:0:0
Credit Point: 2
No. of Lectures: 24

Prerequisites: A basic understanding of employee relations and workforce dynamics in the industry and service sector.

Course Objective(s):

- To equip the students with the basic knowledge and skills of organizational dynamics.
- To equip the students with practical skills in various HRM functions like recruitment, training and performance management.
- To enable students to explore the factors influencing individual and group behavior within organizations.
- Enable students to apply their knowledge to real-world organizational challenges and improve organizational effectiveness.

Course Outcome(s):
After completion of the course students will be able to**:**

CO1: To define and explain the basic concepts of organizational behaviour and motivation.
CO2: To explain the essential concepts of organisational conflicts, resolution of conflicts through negotiation, change management and organisational development.
CO3: To familiarize the various aspects of HR and HR functions in the organization.
CO4: To understand the concepts of HRD, its role and importance in the success of organization.

Course Content:
Module 1: Introduction to Human Resource Management [4L]
HRM: Meaning, definition and functions. Job Analysis, Job Design, Human Resource Planning - Recruitment and Selection - Sources of Recruitment - Selection process, Placement and Induction

Module 2: Human Resource Development [6L]
Introduction to Human Resource Development: Concepts - Training and Development - methods of training, importance of Performance Appraisal, traditional and modern methods of performance appraisal, Job Evaluation - methods of Job Evaluation, Wage and Salary Administration

Module 3: Introduction to Organizational Behaviour [4L]
Meaning and scope of organisational behaviour - Challenges and Opportunities – Foundations of Individual behaviour, Motivation - Theories (Maslow, ERG, Douglas

McGregor two-factor theory), Group dynamics, Leaderships styles. Leadership case studies.

Module 4: Organizational Conflict and Change [5L]
Organizational Conflict - causes and consequences - conflict and negotiation, Organizational change, change management process, resistance to change, flexibility and crisis management – Case studies on organizational change-Organisational Development – concept and significance.

Module 5: Micro and Macro Perspectives on Organizational Behaviour [5L]
Personality, attitudes and values—Understanding motivation performance—Rewards and Behavioral management—Decision Making—Organizational structure, design and culture—Diversity and ethical behaviour in organizations (global perspectives). Gender issues in management. Case studies.

Text book:
1. Simy Joy, Payal Anand, Priya Nair Rajeev. *Organizational Behaviour*. Pearson Education, 2019.
2. K. Aswathappa. *Organizational Behaviour*. 12th edition. Himalaya, 2016.
3.K.Venkataratnam. *Human Resource Management*. 1st edition, Seven Hills Book Publications, 2011.
4. K. Aswathappa, Human Resource Management: Text and Cases. 6th edition, Tata McGraw Hill Education, 2010.

Reference books:
1. M.A. Hitt, C. C. Miller and A. Colella. *Organizational Behaviour: A Strategic Approach*. Hoboken. 2006.
2. G. Johns and A. M. Saks. *Organizational Behavior: Understanding and managing life at work*. 12th edition, Pearson, 2023.
3. Fred Luthans. *Organizational Behaviour*. *8th ed.*Mc Graw –Hill, 1995.
4. Stephen P. Robbins and Timothy A. Judge. *Organizational Behaviour*. 15th ed. Pearson, 2013.

CO–PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	1	2	-	2	3	2	3	1	3	1	2	2
CO2	1	2	2	2	-	2	3	2	3	2	3	1	2	2
CO3	1	2	2	2	-	2	3	2	3	2	3	1	3	3
CO4	1	2	2	2	-	2	3	2	3	2	3	1	3	3

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Course Name: Research Methodology and Intellectual Property Rights
Course Code: HU702
Contact (Periods/Week): 1:0:0
Credit Point: 1
No. of Lectures: 12

Course Objective(s):

- To introduce the fundamentals of research methodology and techniques for identifying research problems.
- To provide awareness on literature review and ethical conduct in research.
- To develop understanding of intellectual property rights (IPR) and its implications in academia and industry.

Course Outcome(s):
After completion of the course, students will be able to:

CO1: Define and formulate a research problem.
CO2: Perform a basic literature review and identify research gaps.
CO3: Demonstrate awareness of ethical practices in research and publication.
CO4: Understand the importance of IPR in safeguarding innovations.

Course Contents:
Module I: Introduction to Research Methodology [2L]

Definition, objectives, and significance of research; types of research; steps in research process; formulating research problem; importance of literature review; primary and secondary sources; identifying research gaps.

Module II: Research Ethics and Integrity [2L]

Research misconduct (Falsification, Fabrication, Plagiarism); conflict of interest; predatory journals; ethical publishing practices; citation practices; tools for plagiarism detection.

Module III: Basics of Report Writing [2L]

Structure of a research report; academic referencing; bibliography; abstracting and summarizing techniques.

Module IV: Intellectual Property Rights [6L]

Introduction to IPR: patents, copyrights, trademarks, GI. Elements of Patentability: Novelty, Non Obviousness (Inventive Steps), Legal requirements for patents — Granting of patent.

Patent application process: Searching a patent- Drawing of a patent- Filing of a patent- Types of patent applications- Patent document: specification and Claims. Govt. Schemes of IPR

Trademarks- Concept of Trademarks - Different kinds of marks (brand names, logos, signatures, symbols, well known marks, certification marks and service marks) - Non Registrable Trademarks - Registration of Trademarks.

Copyrights Right and protection covered by copyright - Law of copy rights: Fundamental of copyright law. Originality of material, rights of reproduction, rights to perform the worth publicly, copy right ownership issues, obtaining copy right registration.

Geographical Indication of Goods, GI Protection.

Textbooks:

- C. R. Kothari – Research Methodology: Methods and Techniques, New Age International.
- Catherine J. Holland – Intellectual Property: Patents, Trademarks, Copyrights, Trade Secrets, Entrepreneur Press, 2007.

Reference Books:

- The Institute of Company Secretaries of India – Professional Programme: Intellectual Property Rights, Law and Practice, Sept 2013.
- Miro Todorovich, Paul Kurtz, Sidney Hook – The Ethics of Teaching and Scientific Research.

CO-PO-PSO Mapping:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	1	-	-	-	-	-	-	2	-			
CO2	3	2	2	-	-	-	-	-	-	2	-			
CO3	2	-	-	-	-	3	3	2	-	2	-			
CO4	2	-	-	-	-	2	3	2	-	2	-			

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

Subject: Rapid Prototyping Lab
Course Code: PR(CS)792
Course Type: Laboratory
Credits: 1.5 (L: 0, T: 0, P: 3)

Prerequisites: Object Oriented Programming through Java, HTML Basics Course

Course Objective(s)

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

1. Apply software engineering and OS knowledge to quickly design and implement working prototypes.
2. Use rapid development tools, frameworks, and libraries for accelerated software creation.
3. Integrate hardware and software components in functional demos (if applicable).
4. Evaluate prototypes based on usability, scalability, and maintainability.
5. Iterate on prototypes based on feedback and testing results.

Course Outcomes (COs)

After completing the course, students will be able to:

CO1: Translate problem requirements into a working prototype within short time frames.
CO2: Select and apply appropriate rapid development frameworks and APIs.
CO3: Integrate multiple system components (UI, backend, database, APIs, hardware if relevant).
CO4: Conduct quick user testing and implement iterative improvements.
CO5: Document and present prototypes effectively for stakeholders.

List of Experiments / Lab Tasks

1. **Introduction to Rapid Prototyping Tools & Platforms** – Overview of Figma, Balsamiq, Streamlit, Flask/Django, Node.js, Android Studio, low-code platforms.
2. **Rapid Web App Prototype** – Build a basic CRUD application in one session using a web framework.
3. **Rapid Mobile App Prototype** – Develop a functional Android/iOS mock app using a cross-platform tool (Flutter/React Native).
4. **API Integration** – Create a prototype consuming an external API (e.g., weather, maps, AI service).
5. **Database-Backed Prototype** – Quick setup using SQLite/PostgreSQL with a simple frontend.

6. **Command-Line Tool Prototype** – Build a utility tool automating a common OS-level task.
7. **Hardware-Software Integration (Optional)** – Rapid prototype using Arduino/ESP32 with a simple Python/JavaScript dashboard.
8. **Minimum Viable Product (MVP) in a Day** – End-to-end prototype for a chosen problem statement.
9. **Usability Testing Session** – Gather feedback from peers and document improvements.
10. **Final Project** – Team-based rapid prototyping challenge on a real-world problem (software-only or hardware-software).

References

1. Todd Zaki Warfel, *Prototyping: A Practitioner's Guide*, Rosenfeld Media.
2. Floyd Mueller et al., *Prototyping Across the Disciplines*, Morgan & Claypool.
3. Bruce Hanington & Bella Martin, *Universal Methods of Design*.
4. Official documentation of selected frameworks (Flask, React, Flutter, etc.).

CO-PO Mapping

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	3	2	1	-	-	-	-	-	-	2	-			
CO2	3	2	2	-	-	-	-	-	-	2	-			
CO3	2	-	-	-	-	3	3	2	-	2	-			
CO4	2	-	-	-	-	2	3	2	-	2	-			

R25 (B. Tech CSE)

Department: Computer Science & Engineering

Curriculum Structure & Syllabus

(Effective from 2025-26 admission batch)

4th Year 8th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credit Points**
					L	**T**	**P**	**Total**	
B. PRACTICAL									
1	PRJ	PRJ	CS881	Internship/Entrepreneurship	0	0	12	12	6
2	ENGG	Major	CS882	Grand Viva	0	0	0	0	2
Total of Theory, Practical and Mandatory Activities/Courses								**12**	**8**
Total Fourth Year Credit									**25**