ASSAM UNIVERSITY, SILCHAR

(A Central University)

Triguna Sen School of Technology

Department of Computer Science and Engineering

Course Structure and Detailed Syllabus

Bachelor of Technology

Computer Science and Engineering

(2026 - 2027 Onwards)



Preface

This booklet contains the Course Structure and Detailed Syllabus of the Bachelor of Technology programme in Computer Science and Engineering offered by the Department of Computer Science and Engineering, Triguna Sen School of Technology, Assam University, Silchar.

The curriculum has been designed in accordance with the National Education Policy (NEP) 2020 and the AICTE Model Curriculum while preserving the approved course contents adopted by the Department.

The syllabus provides comprehensive details regarding:

- Programme Structure
- Credit Distribution
- Semester-wise Course Structure
- Detailed Syllabi
- Course Objectives
- Course Outcomes
- Laboratory Components
- Suggested Text Books

This syllabus shall come into effect from the Academic Session 2027–2028 onwards.

Department of Computer Science and Engineering
Triguna Sen School of Technology
Assam University, Silchar

Contents

Preface **i**

1 Programme Overview **1**

1.1 About Assam University 1

1.2 Department of Computer Science and Engineering 1

1.3 Vision 2

1.4 Mission 2

1.5 Programme Educational Objectives (PEOs) 2

1.6 Programme Outcomes (POs) 3

1.7 Programme Specific Outcomes (PSOs) 4

2 Programme Structure **5**

2.1 Programme Framework 5

2.2 Definition of Credit 5

2.3 Credit Requirement 6

2.4 Code Definitions 6

2.5 Credit Distribution 6

2.6 Composition of Course Types 7

2.7 Student Induction Programme (SIP) 8

2.8 Course Curriculum Structure 8

3 Semester-Wise Detailed Syllabus **15**

1. Programme Overview

About Assam University

Assam University came into existence in 1994 after enactment of the Assam (Central) University Act, 1989. Through its pursuit, Assam University is in the process of making itself an institute of excellence.

The University is situated at Dargakona, approximately 20 km from Silchar, Assam. The picturesque campus extends over nearly 600 acres of land and provides an ideal environment for teaching, learning and research. The University also has another campus at Diphu in the district of Karbi Anglong, Assam.

Assam University has territorial jurisdiction over the districts of Cachar, Karimganj, Hailakandi, Dima Hasao and Karbi Anglong. As a Central University, it reflects the national spirit of unity in diversity by attracting students, faculty and researchers from across India as well as international students.

Department of Computer Science and Engineering

The Department of Computer Science and Engineering (formerly Department of Information Technology) was established during the academic session 2006–2007 under the Triguna Sen School of Technology. It initially started the undergraduate programme in Computer Science and Engineering and subsequently introduced postgraduate and doctoral programmes.

Since its inception, the Department has continuously strengthened its academic and research activities by offering quality education, promoting innovation, interdisciplinary research and industry collaboration. The Department has received research projects funded by various national agencies and has established itself as one of the leading engineering departments in North-East India.

The Department has well-equipped laboratories, experienced faculty members and an outcome-based curriculum designed according to the National Education Policy (NEP) 2020 and AICTE guidelines. The curriculum provides students with strong theoretical foundations, practical exposure and opportunities for interdisciplinary learning, innovation and entrepreneurship.

Graduates of the Department are employed in reputed multinational companies, public sector organizations, startups and research institutions, while many pursue higher studies

in premier national and international universities.

Vision

Vision
To meet the changing needs of the society by: • Building a tradition of innovation. • Solving problem with interdisciplinary collaboration. • Enabling students to discover and apply knowledge for the benefit of society.

Mission

Mission
To meet the changing needs of the society by: • To educate the students in the areas of Computer Science & Engineering by providing best practices of teaching-learning process for careers in industries, higher education and research. • To provide the students with foundation of both the theory and practice through inclusive undergraduate and postgraduate curriculum, and strengthening them for creativity and innovation for serving the society.

Programme Educational Objectives (PEOs)

Graduates of the B.Tech. (CSE) programme are expected to:

PEO	Description
PEO 1	Will demonstrate successful professional career by impacting cutting edge technologies to accomplish social responsibilities.
PEO 2	Will demonstrate lifelong learning, critical thinking to extract innovative technical solutions using research-oriented, complex problem-solving skills.
PEO 3	Will demonstrate professionalism, ethical behaviour, inter-personal skills to develop leadership qualities.

Programme Outcomes (POs)

The graduates of the programme will be able to:

PO	Description
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of their limitations.
PO6	The engineer and society: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.
PO7	Environment and sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
PO9	Individual and team work: Function effectively as an individual and as a member or leader in diverse teams and multidisciplinary environments.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and society at large through reports, documentation, presentations and effective interaction.
PO11	Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to individual and team-based projects.
PO12	Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs)

At the time of graduation, students will be able to:

PSO	Description
PSO1	Academic Competence: Apply various concepts of Computing, Statistics, Mathematics and Electronics appropriately to conduct research as well as to provide solutions to real-life societal problems.
PSO2	Professional Competence: Develop strong problem-solving abilities for professional software development as well as to analyse critical problems.
PSO3	Entrepreneurial Competence: Acquire the skills necessary to support careers in software development, entrepreneurship and emerging areas such as data analytics, artificial intelligence, healthcare imaging, networking and embedded systems.

2. Programme Structure

Programme Framework

The Bachelor of Technology (B.Tech.) programme in Computer Science and Engineering has been designed in accordance with the National Education Policy (NEP) 2020 and the AICTE Model Curriculum. The curriculum is structured to provide a balanced blend of fundamental sciences, engineering sciences, professional core courses, electives, project-based learning and multidisciplinary education.

The salient features of the programme are as follows:

- Minimum credit requirement for completion of the programme is **170 credits**.
- The curriculum is based on the guidelines of National Education Policy (NEP) 2020 and AICTE.
- The curriculum is designed to meet prevailing and emerging industrial requirements.
- The curriculum incorporates project-based education with projects distributed across different years of study.
- The curriculum follows the Choice Based Credit System (CBCS).
- The curriculum includes value-based education, Indian Knowledge System (IKS) and interdisciplinary learning.
- The curriculum encourages digital pedagogy, innovation, entrepreneurship and startup culture.
- Students are encouraged to enroll in NPTEL/MOOC courses subject to the prevailing regulations of the Department.
- The curriculum aims at the holistic development of students.

Definition of Credit

Activity	**Credit**
1 Hour Lecture (L) per week	1
1 Hour Tutorial (T) per week	1
2 Hours Practical/Laboratory (P) per week	1

Credit Requirement

A student is required to successfully complete a minimum of **170 credits** for the award of the Bachelor of Technology degree in Computer Science and Engineering. These credits are earned through theory courses, laboratory courses, professional electives, open electives, project work, internship and other curriculum components prescribed by the Department.

Code Definitions

Sl. No.	Code	Code Definition
1	HC	Humanities and Social Sciences including Management Courses
2	BC	Basic and Applied Science Courses
3	EC	Engineering Science Courses including Workshop, Drawing and Basic Engineering Courses
4	PC	Professional Core Courses
5	PE	Professional Elective Courses
6	OE	Open Elective Courses
7	PR	Project Work, Seminar, Grand Viva and Internship
8	MC	Mandatory Courses
9	L	Lecture
10	T	Tutorial
11	P	Practical
12	C	Credits

Credit Distribution

Sl. No.	Category	Credits
1	Humanities and Social Sciences including Management Courses	15
2	Basic and Applied Science Courses	20
3	Engineering Science Courses	28
4	Professional Core Courses	67
5	Professional Elective Courses	15
6	Open Elective Courses	09
7	Project Work, Seminar, Grand Viva and Internship	16
8	Mandatory Courses	No Credits
	Total Credits	**170**

Composition of Course Types

Sl. No.	Category	Credits	Percentage
1	Humanities and Social Sciences including Management Courses	15	8.82
2	Basic and Applied Science Courses	20	11.76
3	Engineering Science Courses	28	16.47
4	Professional Core Courses	67	39.41
5	Professional Elective Courses	15	8.82
6	Open Elective Courses	09	5.29
7	Project Work, Seminar, Grand Viva and Internship	16	9.41
8	Mandatory Courses	0	0.00
	Total	**170**	**100**

Student Induction Programme (SIP)

Student Induction Programme (SIP)

As per the latest AICTE directives, the Student Induction Programme (SIP) aims to:

1. Become familiar with the ethos and culture of the institution.
2. Develop a healthy daily routine and foster bonding among peers and faculty members.
3. Gain exposure to a holistic vision of life and develop awareness of the Self, family, society, nation, and nature.
4. Build meaningful relationships with peers and seniors that extend beyond academic life.
5. Strengthen essential professional skills (e.g., Mathematics, Language proficiency) for students who require additional support.

Programme Highlights:

- **Duration:** Two - Three weeks
- Conducted at the beginning of the first year
- Physical Activities and Creative Arts
- Universal Human Values
- Literary Activities
- Proficiency Modules
- Lectures by Eminent Speakers
- Visits to Local Areas
- Familiarization with Department/Branch and Innovations

Course Curriculum Structure

The curriculum provides adequate flexibility through elective courses, enabling students to pursue specialised areas of interest while maintaining a strong foundation in core computer science and engineering principles. In addition, value-added components such as internships, community engagement, innovation, entrepreneurship, and multidisciplinary learning opportunities are incorporated to enhance employability and foster lifelong learning. The semester-wise distribution of courses, contact hours, and credits is presented in the subsequent sections.

Student Induction Programme (SIP) :

Student Induction Programme (SIP)

As per the latest AICTE directives, the Student Induction Programme (SIP) aims to:

1. Become familiar with the ethos and culture of the institution.
2. Develop a healthy daily routine and foster bonding among peers and faculty members.
3. Gain exposure to a holistic vision of life and develop awareness of the Self, family, society, nation, and nature.
4. Build meaningful relationships with peers and seniors that extend beyond academic life.
5. Strengthen essential professional skills (e.g., Mathematics, Language proficiency) for students who require additional support.

Programme Highlights:

- **Duration:** Two - Three weeks
- Conducted at the beginning of the first year
- Physical Activities and Creative Arts
- Universal Human Values
- Literary Activities
- Proficiency Modules
- Lectures by Eminent Speakers
- Visits to Local Areas
- Familiarization with Department/Branch and Innovations

Semester-wise Structure of Curriculum

First Year - 1st Semester

Mandatory Induction Programme							
Sl. No.	**Course Type**	**Course Code**	**Course Title**	**Hours/Week**			**Credits**
				L	**T**	**P**	
1	BC	ASH101	Physics-I	3	1	0	4
2	BC	ASH102	Mathematics–I	3	1	0	4
3	EC	ASH104	Basic Electrical Engineering	3	1	0	4
4	BC	ASH105	Physics-I Lab	0	0	2	1
5	EC	ASH107	Engineering Graphics and Design Thinking	0	0	6	3
6	EC	ASH108	Basic Electrical Engineering Lab	0	0	2	1
7	MC (Audit)	ASH109	IDEA Lab Workshop	0	0	6	0
			Total Credits				**17**

First Year - 2nd Semester

Sl. No.	**Course Type**	**Course Code**	**Course Title**	**Hours/Week**			**Credits**
				L	**T**	**P**	
1	BC	ASH202	Mathematics–II	3	1	0	4
2	BC	ASH203	Chemistry-I	3	1	0	4
3	HC	ASH204	English	3	0	0	3
4	BC	ASH206	Chemistry-I Lab	0	0	2	1
5	EC	ASH207	Programming for Problem Solving	3	0	0	3
6	HC	ASH208	Universal Human Values	2	1	0	3
7	EC	ASH209	Programming for Problem Solving Lab	0	0	4	2
8	EC	ASH210	Workshop/Manufacturing Practices	0	0	6	3
			Total Credits				**23**

Semester-wise Structure of Curriculum

Second Year - 3rd Semester

Sl. No.	Course Type	Course Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE301	Discrete Mathematics & Graph Theory	3	1	0	4
2	PC	CSE302	Data Structures	3	0	0	3
3	EC	CSE303	Digital Electronics	3	0	0	3
4	PC	CSE304	Object Oriented Programming	3	0	0	3
5	PC	CSE305	Data Structures Lab	0	0	4	2
6	EC	CSE306	Digital Electronics Lab	0	0	4	2
7	PC	CSE307	Object Oriented Programming Lab	0	0	4	2
8	BC	ASH301	Mathematics–III	2	0	0	2
9	HC	ASH302	Effective Technical Communication	3	0	0	3
			Total Credits				**24**

Second Year - 4th Semester

Sl. No.	Course Type	Course Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE401	Formal Languages and Automata Theory	3	1	0	4
2	PC	CSE402	Data Science Using Python	3	0	0	3
3	PC	CSE403	Database Management Systems	3	0	0	3
4	PC	CSE404	Computer Organization and Architecture	3	1	0	4
5	PC	CSE405	Data Science Using Python Lab	0	0	4	2
6	PC	CSE406	Database Management Systems Lab	0	0	4	2
7	PC	CSE407	IT Workshop I (Web Development)	0	0	4	2
8	HC	ASH401	Management-I (Organisational Behaviour)	3	0	0	3
9	MC (Audit)	ASH402	Environmental Science	2	0	0	0
			Total Credits				**23**

Note: Students are required to undertake a mandatory **Summer Training Programme** of at least 4 weeks (40 hours) in duration at reputed industries, organizations, or research institutes after the completion of the 4th semester.

Semester-wise Structure of Curriculum

Third Year - 5th Semester

Sl. No.	Course Type	Course Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE501	Software Engineering	3	1	0	4
2	PC	CSE502	Design and Analysis of Algorithms	3	0	0	3
3	PC	CSE503	Operating System	3	0	0	3
4	PC	CSE504	Computer Networks	3	0	0	3
5	PC	CSE505	Design and Analysis of Algorithms Lab	0	0	4	2
6	PC	CSE506	Operating System Lab	0	0	4	2
7	PC	CSE507	Computer Networks Lab	0	0	4	2
8	PR	CSE508	Summer Training (4 Weeks)	0	0	0	0
9	HC	ASH501	Management II (Project Management / Operations Research and Industrial Management)	3	0	0	3
10	MC	ASH502	Essence of Indian Knowledge Tradition/-Constitution of India	2	0	0	0
			Total Credits				**22**

Third Year - 6th Semester

Sl. No.	Course Type	Course Code	Course Title	Hours/Week			Credits
				L	T	P	
1	EC	CSE601	Research Methodology and Technical Writing	3	0	0	3
2	PC	CSE602	Compiler Design	3	0	0	3
3	PC	CSE603	Machine Learning	3	0	0	3
4	PE	CSE604(#)	Elective I	3	0	0	3
5	OE	CSE605(#)	Open Elective I	3	0	0	3
6	PC	CSE606	Compiler Design Lab	0	0	4	2
7	PC	CSE607	Machine Learning Lab	0	0	4	2
8	PC	CSE608	IT Workshop II (Mobile Application Development)	0	0	4	2
9	PR	CSE609	Minor Project	0	0	6	3
			Total Credits				**24**

Note: Students will have to take up **Mandatory Internship/Apprenticeship** of 8 weeks (80 hrs) at industry/organizations/research institutes of repute after the completion of 6th semester, that will be accredited in 7th semester.

Semester-wise Structure of Curriculum

Fourth Year - 7th Semester

Sl. No.	Course Type	Course Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE701	Cryptography and Network Security	3	0	0	3
2	PC	CSE702	Deep Learning	3	0	0	3
3	PE	CSE703(#)	Elective II	3	0	0	3
4	PE	CSE704(#)	Elective III	3	0	0	3
5	OE	CSE705(#)	Open Elective II	3	0	0	3
6	PR	CSE706	Internship (8 weeks)	0	0	0	2
7	PR	CSE707	Major Project I	0	0	8	4
			Total Credits				**21**

Fourth Year - 8th Semester

Sl. No.	Course Type	Course Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PE	CSE801(#)	Elective IV	3	0	0	3
2	PE	CSE802(#)	Elective V	3	0	0	3
3	OE	CSE803(#)	Open Elective III	3	0	0	3
4	PR	CSE804	Grand Viva	0	0	2	1
5	PR	CSE805	Major Project II	0	0	12	6
			Total Credits				**16**

Note: # denotes the serial number of the elective, as listed under the Professional Electives or Open Electives, as applicable.

Note: Students selecting open electives offered by other departments are required to obtain the appropriate course codes directly from the respective departments offering those courses, to ensure accurate registration and documentation.

List of Electives

Professional Electives

No.	Elective	No.	Elective
1	Advanced Algorithms	21	Introduction to Large Language Models
2	Advanced Computer Architecture	22	Mobile Computing
3	Advanced DBMS	23	Natural Language Processing
4	Advanced Operating System	24	Network-On-Chip
5	Advanced Software Development	25	Distributed Systems
6	Artificial Intelligence	26	Quantum Computing
7	Bioinformatics	27	Recommendation Systems
8	Blockchain	28	Reinforcement Learning
9	Cloud Computing	29	Robotics
10	Computational Complexity	30	Social Network Analysis
11	Computational Geometry	31	Soft Computing
12	Computer Graphics	32	Software Testing
13	Computer Vision	33	Speech Processing
14	Cyber Security	34	User Interface Engineering
15	Data Analytics	35	Evolutionary Algorithms
16	Data Mining	36	Formal Methods for System Verification
17	Digital Forensics	37	Full Stack Development
18	Internet of Things	38	Game Development
19	Green Computing	39	Human Computer Interaction
20	Image Processing	40	Information Retrieval

Open Electives

S.No.	Open Elective Course	S.No.	Open Elective Course
1	Fundamentals of Machine Learning	6	Technical Writing
2	Data Analytics	7	Soft Computing
3	Data Science	8	Software Engineering
4	MATLAB	9	UI/UX Design Fundamentals
5	Python Programming	10	Web and Internet

Note: The curriculum also allows students to undertake NPTEL courses as elective subjects, provided they choose from the list of courses notified by the respective department(s) from time to time, and are evaluated in accordance with the prevailing norms of the Department of CSE.

3. Semester-Wise Detailed Syllabus

Programme Overview

The curriculum of the B.Tech. programme in Computer Science and Engineering is designed as a progressive learning pathway that integrates fundamental engineering knowledge, mathematical foundations, computer science theory, professional core competencies, laboratory experience, electives, industrial exposure, research orientation, and project-based learning.

The curriculum progresses from foundational concepts in the early semesters to advanced computing concepts, professional electives, open electives, industrial exposure, and major project work in the later semesters. The structure is intended to provide students with a balanced combination of **theoretical knowledge, practical skills, problem-solving ability, professional competence, and independent learning**.

Major Curriculum Components

FOUNDATION	**COMPUTING**
Mathematics and Science	Computing Foundation
Basic Engineering	Professional Core
Humanities and Social Sciences	Laboratory and Practical Learning
PROFESSIONAL DEVELOPMENT	**ADVANCED LEARNING**
Research and Innovation	Professional Electives
Industrial Exposure	Open Electives
Project-Based Learning	Advanced and Emerging Computing
Professional and Holistic Development	Independent and Applied Learning

DETAILED SYLLABUS

First Year – 1st Semester

Mandatory Induction Programme							
Sl.	**Type**	**Code**	**Course Title**	**Hours/Week**			**Credits**
				L	**T**	**P**	
1	BC	ASH101	Physics-I	3	1	0	4
2	BC	ASH102	Mathematics–I	3	1	0	4
3	EC	ASH104	Basic Electrical Engineering	3	1	0	4
4	BC	ASH105	Physics-I Lab	0	0	2	1
5	EC	ASH107	Engineering Graphics and Design Thinking	0	0	6	3
6	EC	ASH108	Basic Electrical Engineering Lab	0	0	2	1
7	MC	ASH109	IDEA Lab Workshop	0	0	6	0
						Total Credits	**17**

ASH101 – Physics-I

Course Information	
Course Code	ASH101
Course Name	Physics-I
Category	Basic Science Course
L–T–P–C	3–1–0–4

Course Objectives

1. To use scalar and vector analytical techniques for analysing forces
2. To understand basic kinematics concepts – displacement, velocity and acceleration (and their angular counterparts);
3. To study Bragg's Law and introduce the basic concept of crystallography
4. To study the basic concepts of quantum physics.
5. To understand the principles of semiconductor Physics
6. Physics lab provides students with the first-hand experience of verifying various theoretical concepts learnt in theory courses.

Course Outcomes

At the end of the course, the students will be able to learn the basics of physics and apply them to solve engineering problems.

1. Understand and be able to apply Newton's laws of motion.
2. Understand and be able to apply other basic dynamics concepts - the Work-Energy principle and Impulse-Momentum.
3. Knowledge to solve simple quantum mechanics calculations
4. Understand and utilize the mathematical models of semiconductor junctions
5. Understand various laws which they have studied through experiments
6. Apply basic knowledge of physics to solve real-world problems

Detailed Syllabus

Unit	Course Contents
UNIT I	Transformation of scalars and vectors under Rotation transformation; Forces in Nature; Newton's laws and its completeness in describing particle motion; Solving Newton's equations of motion in polar coordinates; Potential energy function; $F = -\text{Grad}\, V$; Conservative and non-conservative forces; Central forces; Conservation of Angular Momentum; Energy equation and energy diagrams; Elliptical, parabolic and hyperbolic orbits; Application: Satellite manoeuvres.
UNIT II	Crystal structure: Seven systems of crystals, Bravais space lattice, crystal structure (bcc, fcc and sc) lattice dimensions, lattice planes, and miller indices and their significance, X-rays-absorption of X-rays diffraction, Bragg's law. Bragg's X-ray spectrometer.
UNIT III	Wave particle duality, Uncertainty principle, Free-particle wave function and wave-packets, probability current, Expectation values, Schrodinger equation and its application to particle in a box and harmonic oscillator, Numerical solution of stationary-state Schrodinger equation for one dimensional problems for different potentials.
UNIT IV	Harmonic oscillator; damped harmonic oscillator – heavy, critical and light damping, energy decay in a damped harmonic oscillator, quality factor, forced mechanical and electrical oscillators, electrical and mechanical impedance, steady state motion of forced damped harmonic oscillator, power absorbed by oscillator.
UNIT V	Intrinsic and extrinsic semiconductors, Dependence of Fermi level on carrier-concentration and temperature (equilibrium carrier statistics), Carrier generation and recombination, Carrier transport: diffusion and drift, p-n junction, Metal-semiconductor junction (Ohmic and Schottky), Semiconductor materials of interest for optoelectronic devices.

Suggested Books

References
Engineering Mechanics, 2nd ed. — MK Harbola
Introduction to Mechanics — MK Verma
An Introduction to Mechanics — D Kleppner & R Kolenkow
Principles of Mechanics — JL Synge & BA Griffiths
Mechanics — JP Den Hartog
Introduction to Quantum Physics D. J. Griffiths
J. Singh, Semiconductor Optoelectronics: Physics and Technology, McGraw-Hill Inc. (1995).
B. E. A. Saleh and M. C. Teich, Fundamentals of Photonics, John Wiley & Sons, Inc., (2007).
S. M. Sze, Semiconductor Devices: Physics and Technology, Wiley (2008).
Charks Kittle, Introduction to Solid State Physics, John Wiley & Sons

Chottopadhyay and Rakshit, Quantum Mechanics, Statistical Mechanics and Solid State Physics
Theory of Vibrations with Applications — WT Thomson

ASH102 – Mathematics-I (Calculus and Linear Algebra)

Course Information	
Course Code	ASH102
Course Name	Mathematics-I
Category	Basic Science Courses
L–T–P–C	3–1–0–4
Pre-requisites	Basic knowledge of +2 level calculus and algebra

Course Objectives
1. To introduce the idea of applying differential and integral calculus to the notions of Curvature and to improper integrals. Apart from some engineering applications, it gives a basic introduction on Beta and Gamma functions. 2. To discuss Mean Value Theorems that is fundamental to application of analysis to Engineering problems. 3. To develop the tool of matrices to solve systems of linear equations arising in many engineering problems by different methods. 4. To familiarize the students with the concepts of vector spaces that is essential in most branches of engineering.

Course Outcomes
At the end of the course the students should be able to 1. understand the basic knowledge of Calculus and its applications 2. be familiar with the concept of Matrices and solution of system of linear equations 3. be thorough with the concept of Linear Algebra and its applications in engineering

Detailed Syllabus

Unit	Course Contents
UNIT I	Differential Calculus: Rolle's Theorem; Mean value theorems; indeterminate forms and L'Hospital's rule; Maxima and minima.
UNIT II	Integral Calculus: Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate surface areas and volumes of revolutions.
UNIT III	Matrices: Matrices, vectors: addition and scalar multiplication, matrix multiplication; Linear systems of equations, linear independence, rank of a matrix, determinants.

Unit	Course Contents
UNIT IV	Vector Spaces and Linear Transformation: Vector Space, linear dependence and independence of vectors, basis, dimension; Linear transformations (maps), range and kernel of a linear map, rank and nullity, rank-nullity theorem, Matrix associated with a linear map.
UNIT V	Eigenvalues, eigenvectors and diagonalization: Eigenvalues, eigenvectors. Diagonalization. Their applications in the solution of system of linear equations.

Text Books

Text Books
1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002. (Unit-I, II)
2. E. Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006. (Unit-III, IV, V)

References

References
1. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
2. D. Poole, Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005.
3. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
4. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.

ASH104 – Basic Electrical Engineering

Course Information	
Course Code	ASH104
Course Name	Basic Electrical Engineering
Category	Basic Science Course
L–T–P–C	3–1–0–4

Course Objectives
17. To understand and analyze basic electric and magnetic circuits.
18. To study the working principles of electrical machines and power converters.
19. To introduce the components of high and low-voltage electrical installations.

Course Outcomes
The students will learn:
1. To explain the strong basics of Electrical Engineering and practical implementation of Electrical fundamentals.
2. To identify different applications of commonly used electrical machinery.

Detailed Syllabus

Unit	Course Contents
UNIT I	**D. C. Circuits** covering, Ohm's Law and Kirchhoff's Laws; Analysis of series, parallel and series-parallel circuits excited by independent voltage sources; Power and energy; **Electromagnetism** covering, Faradays Laws, Lenz's Law, Fleming's Rules, Statically and dynamically induced EMF; Concepts of self-inductance, mutual inductance and coefficient of coupling; Energy stored in magnetic fields.
UNIT II	**Single Phase A.C. Circuits** covering, Generation of sinusoidal voltage-definition of average value, root mean square value, form factor and peak factor of sinusoidal voltage and current and phasor representation of alternating quantities; Analysis with phasor diagrams of R, L, C, RL, RC and RLC circuits; Real power, reactive power, apparent power and power factor, series, parallel and series-parallel circuits; **Three Phase A.C. Circuits** covering, Necessity and Advantages of three-phase systems, Generation of three-phase power, definition of Phase sequence, balanced supply and balanced load; Relationship between line and phase values of balanced star and delta connections; Power in balanced three-phase circuits, measurement of power by two wattmeter method.

Unit	Course Contents
UNIT III	**Transformers** covering, Principle of operation and construction of single-phase transformers (core and shell types). EMF equation, losses, efficiency and voltage regulation; **Synchronous Generators** covering, Principle of operation; Types and constructional features; EMF equation.
UNIT IV	**DC Machines** covering, working principle of DC machine as a generator and a motor; Types and constructional features; EMF equation of generator, relation between EMF induced and terminal voltage enumerating the brush drop and drop due to armature reaction; DC motor working principle; Back EMF and its significance, torque equation; Types of D.C. motors, characteristics and applications; Necessity of a starter for DC motor.
UNIT V	**Three Phase Induction Motors** covering; Concept of rotating magnetic field; Principle of operation, types and constructional features; Slip and its significance; Applications of squirrel cage and slip ring motors; Necessity of a starter, star-delta starter. **Sources of Electrical Power** covering, Introduction to Wind, Solar, Fuel cell, Tidal, Geothermal, Hydroelectric, Thermal-steam, diesel, gas, nuclear power plants; Concept of cogeneration, and distributed generation.

Suggested Books

References
1. AICTE's Prescribed Textbook: Basic Electrical Engineering, Khanna Book Publishing.
2. Ritu Sahdev (2022), Basic Electrical Engineering, Khanna Book Publishing.
3. Nagrath I.J. and D. P. Kothari (2001), Basic Electrical Engineering, Tata McGraw Hill.
4. Hayt and Kimberly, Engineering Circuit Analysis, Tata McGraw Hill.
5. Kulshreshtha D.C. (2009), Basic Electrical Engineering, Tata McGraw Hill.
6. Rajendra Prasad (2009), Fundamentals of Electrical Engineering, Prentice Hall, India Hughes, E. (2005)
7. Mittel & Mittal, Basic Electrical Engineering, Tata McGraw Hill.

ASH105 – Physics-I Lab

Course Information	
Course Code	ASH105
Course Name	Physics-I Lab
Category	Basic Science Course
L–T–P–C	0–0–2–1

Physics Laboratory [L : 0; T : 0; P : 2 (1 credit)]
Experiments from the following:

Experiment Group	Experiments
Introduction to Electromagnetic Theory	1. Magnetic field from Helmholtz coil 2. Measurement of Lorentz force in a vacuum tube.
Introduction to Mechanics	1. Coupled oscillators 2. Experiments on an air-track 3. Experiment on the moment of inertia measurement 4. Experiments with gyroscope 5. Resonance phenomena in mechanical oscillators.
Quantum Mechanics for Engineers	1. Frank-Hertz experiment 2. Photoelectric effect experiment 3. Recording hydrogen atom Spectrum
Oscillations, Waves and Optics	1. Diffraction and interference experiments (from ordinary light or laser pointers) 2. Measurement of the speed of light on a tabletop using modulation 3. Minimum deviation from a prism.

ASH107 – Engineering Graphics & Design Thinking

Course Information	
Course Code	ASH107
Course Name	Engineering Graphics & Design Thinking
Category	Engineering Science Course
L–T–P–C	1–0–4–3

Course Objectives

The objective of this Course is to provide the basic knowledge about Engineering Drawing and to provide the new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products. Detailed concepts are given in projections, technical drawing, dimensioning, designing and specifications, so useful for a student in preparing for an engineering career.

Course Outcomes

After the completion of the course, the learner will be able to

1. Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

2. Use the techniques, skills, and modern engineering tools necessary for engineering practice.

3. Describe engineering design and its place in society.

4. Discuss the visual aspects of engineering design.

5. Use engineering graphics standards.

6. Illustrate solid modelling.

7. Use computer-aided geometric design.

8. Design creating working drawings.

10. Inspect engineering communication.

11. Compare and classify the various learning styles and memory techniques and apply them in their engineering education.

12. Analyze emotional experience and Inspect emotional expressions to better understand users while designing innovative products.

13. Develop new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products.

14. Propose real-time innovative engineering product designs and Choose appropriate frameworks, strategies, techniques during prototype development, Perceive individual differences and its impact on everyday decisions and further Create a better customer experience.

Detailed Syllabus

Unit	Course Contents
Traditional Engineering Graphics: 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. **Computer Graphics:** Engineering Graphics Software; Spatial Transformations; Orthographic Projections; Model Viewing; Co-ordinate Systems; Multi-view Projection; Exploded Assembly; Model Viewing; Animation; Spatial Manipulation; Surface Modelling; Solid Modelling; Introduction to Building Information Modelling (BIM).	
UNIT I	**Introduction to Engineering Drawing and an Insight to Learning Design** Principles of Engineering Graphics and their significance, usage of Drawing instruments, lettering, Conic sections including the Rectangular Hyperbola (General method only); Cycloid, Epicycloid, Hypocycloid and Involute; Scales – Plain, Diagonal and Vernier Scales; Orthographic Projections Principles of Orthographic Projections-Conventions - Projections of Points and lines inclined to both planes; Projections of planes inclined Planes - Auxiliary Planes; Understanding the Learning Process, Kolb's Learning Styles, Assessing and Interpreting, Understanding the Memory process, Problems in retention, Memory enhancement techniques, Understanding Emotions, Experience & Expression, Assessing Empathy, Application with Peers.
UNIT II	**Projections of Regular Solids and Basics of Design Thinking** Covering those inclined to both the Planes- Auxiliary Views; Draw simple annotation, dimensioning and scale. Floor plans that include: windows, doors, and fixtures such as WC, bath, sink, shower, etc. Sections and Sectional Views of Right Angular Solids. Prism, Cylinder, Pyramid, Cone – Auxiliary Views; Development of surfaces of Right Regular Solids - Prism, Pyramid, Cylinder and Cone; Draw the sectional orthographic views of geometrical solids, objects from industry and dwellings (foundation to slab only). Definition of Design Thinking, Need for Design Thinking, Objective of Design Thinking, Concepts & Brainstorming, Stages of Design Thinking Process (explain with examples) – Empathize, Define, Ideate, Prototype, Test.
UNIT III	**Isometric Projections and Being Ingenious & Fixing Problem** Principles of Isometric projection – Isometric Scale, Isometric Views, Conventions; Isometric Views of lines, Planes, Simple and compound Solids; Conversion of Isometric Views to Orthographic Views and Vice-versa, Conventions; Understanding Creative thinking process, Understanding Problem Solving, Testing Creative Problem Solving, Understanding Individual differences & Uniqueness.

Unit	Course Contents
UNIT IV	**Orthographic Projections of Points and Process of Product Design** Orthographic Projections of Points: Introduction to drawing standards, creation of 2D environment using CAD software, principles of orthographic projections, projections of points in all the four quadrants. Process of Engineering Product Design, Design Thinking Approach, Stages of Product Design, Examples of best product designs and functions, Assignment – Engineering Product Design.
UNIT V	**Transformation of Projections and Prototyping & Testing** Conversion of Isometric Views to Orthographic Views. Conversion of orthographic views to isometric views – simple objects. Plan and elevation of simple buildings with dimensions. What is Prototype? Why Prototype? Rapid Prototype Development process, Testing, Sample Example, Test Group Marketing.

Text Books

Text Books
1. Bhatt N.D., Panchal V.M. & Ingle P.R., (2014), Engineering Drawing, Charotar Publishing House.
2. Shah, M.B. & Rana B.C. (2008), Engineering Drawing and Computer Graphics, Pearson Education.
3. Agrawal B. & Agrawal C.M. (2012), Engineering Graphics, TMH Publication.
4. Narayana, K.L. & P. Kannaiah (2008), Text Book on Engineering Drawing, Scitech Publishers.

Text Books / References

As provided by the course instructor.

ASH108 – Basic Electrical Engineering Lab

Course Information	
Course Code	ASH108
Course Name	Basic Electrical Engineering Lab
Category	Basic Science Course
L–T–P–C	0–0–2–1

Basic Electrical Engineering Laboratory [L : 0; T : 0; P : 2 (1 credit)]
List of Experiments/Demonstrations

No.	**Experiment**
1	Basic safety precautions. Introduction and use of measuring instruments–voltmeter, ammeter, multi-meter, oscilloscope. Real-life resistors, capacitors and inductors.
2	Measuring the steady-state and transient time-response of R-L, R-C, and R-L-C circuits to a step change in voltage (transient may be observed on a storage oscilloscope). Sinusoidal steady state response of R-L, and R-C circuits – impedance calculation and verification. Observation of phase differences between current and voltage. Resonance in R-L-C circuits.
3	Transformers: Observation of the no-load current waveform on an oscilloscope (nonsinusoidal wave-shape due to B-H curve nonlinearity should be shown along with a discussion about harmonics). Loading of a transformer: measurement of primary and secondary voltages and currents, and power.
4	Three-phase transformers: Star and Delta connections. Voltage and Current relationships (line-line voltage, phase-to-neutral voltage, line and phase currents). Phase-shifts between the primary and secondary sides. Cumulative three-phase power in balanced three-phase circuits.
5	Demonstration of cut-out sections of machines: dc machine (commutator-brush arrangement), induction machine (squirrel cage rotor), synchronous machine (field winding - slip ring arrangement) and single-phase induction machine.
6	Torque Speed Characteristic of separately excited dc motor.
7	Synchronous speed of two and four-pole, three-phase induction motors. Direction reversal by change of phase-sequence of connections. Torque-Slip Characteristic of an induction motor. Generator operation of an induction machine driven at super-synchronous speed.
8	Synchronous Machine operating as a generator: stand-alone operation with a load. Control of voltage through field excitation.
9	Demonstration of (a) dc-dc converters (b) dc-ac converters – PWM waveform (c) the use of dc-ac converter for speed control of an induction motor and (d) Components of LT switchgear.

ASH109 – IDEA Lab Workshop

Course Information	
Course Code	ASH109
Course Name	IDEA Lab Workshop
Category	Audit Course
L–T–P–C	2–0–4–0
Pre-requisite	None

Course Objectives

The objective of Idea Lab is to make engineering graduates more imaginative and creative in critical thinking, problem solving, design thinking, collaboration, communication, lifelong learning etc. In addition, it will help them

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

Course Outcomes

Upon completion of this laboratory course, students will be able to

1. Study and practice on machine tools and their operations.
2. Practice on manufacturing of components using workshop trades including plumbing, fitting, carpentry, foundry, house wiring and welding.
3. Identify and apply suitable tools for different trades of Engineering processes including drilling, material removing, measuring, chiselling.

Detailed Syllabus

Unit	Course Contents
UNIT I	Introduction to basic hand tools - Tape measure, combination square, Vernier caliper, 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. Various types of drill bits.

Unit	Course Contents
UNIT II	Mechanical cutting processes - 3-axis CNC routing, basic turning, milling, drilling and grinding operations, Laser cutting, Laser engraving etc. Basic welding and brazing and other joining techniques for assembly. Concept of Lab aboard a Box.
UNIT III	3D printing and prototyping technology – 3D printing using FDM, SLS and SLA. Basics of 3D scanning, point cloud data generation for reverse engineering. Prototyping using subtractive cutting processes. 2D and 3D Structures for prototype building using Laser cutter and CNC routers. Basics of IPR and patents; Accessing and utilizing patent information in IDEA Lab.
UNIT IV	Discussion and implementation of a mini project.
UNIT V	Documentation of the mini project (Report and video).

List of Laboratory Activities and Experiments

No.	Activity / Experiment
1	Machining of 3D geometry on soft material such as soft wood or modelling wax.
2	2D profile cutting on plywood / MDF (6–12 mm) for press fit designs.
3	Familiarity and use of welding equipment.
4	Familiarity and use of normal and wood lathe.
5	Design and implementation of a capstone project involving embedded hardware, software and machined or 3D printed enclosure.

Text Book

Text Book
1. The Big Book of Maker Skills: Tools & Techniques for Building Great Tech Projects. Chris Hackett. Weldon Owen; 2018. ISBN-13: 978-1681884325.

References

References
1. The Total Inventors Manual (Popular Science): Transform Your Idea into a Top-Selling Product. Sean Michael Ragan. Weldon Owen; 2017. ISBN-13: 978-1681881584.
2. Make: Tools: How They Work and How to Use Them. Charles Platt. Shroff/Maker Media. 2018. ISBN-13: 978-9352137374.

3. The Art of Electronics. 3rd Edition. Paul Horowitz and Winfield Hill. Cambridge University Press. ISBN: 9780521809269.
4. Practical Electronics for Inventors. 4th Edition. Paul Sherz and Simon Monk. McGraw Hill. ISBN-13: 978-1259587542.
5. Encyclopedia of Electronic Components (Volumes 1, 2 and 3). Charles Platt. Shroff Publishers. ISBN-13: 978-9352131945, 978-9352131952, 978-9352133703.
6. Building Scientific Apparatus. 4th Edition. John H. Moore, Christopher C. Davis, Michael A. Coplan, Sandra C. Greer. Cambridge University Press. ISBN-13: 978-0521878586.
7. Programming Arduino: Getting Started with Sketches. 2nd Edition. Simon Monk. McGraw Hill. ISBN-13: 978-1259641633.
8. Make Your Own PCBs with EAGLE: From Schematic Designs to Finished Boards. Simon Monk and Duncan Amos. McGraw Hill Education. ISBN-13: 978-1260019193.
9. Pro GIT. 2nd Edition. Scott Chacon and Ben Straub. Apress. ISBN-13: 978-1484200773.
10. Venu Vinod, P.K., M.A. W., Rapid Prototyping – Laser Based and Other Technologies, Kluwer, 2004.
11. Ian Gibson, David W. Rosen, Brent Stucker, Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing, Springer, 2010.
12. Chapman W.A.J., Workshop Technology, Volumes I, II and III, CBS Publishers and Distributors, 5th Edition, 2002.

DETAILED SYLLABUS
First Year – 2nd Semester

Sl.	Type	Code	Course Title	Hours/Week			Credits
				L	T	P	
1	BC	ASH202	Mathematics–II	3	1	0	4
2	BC	ASH203	Chemistry-I	3	1	0	4
3	HC	ASH204	English	3	0	0	3
4	BC	ASH206	Chemistry-I Lab	0	0	2	1
5	EC	ASH207	Programming for Problem Solving	3	0	0	3
6	HC	ASH208	Universal Human Values	2	1	0	3
7	EC	ASH209	Programming for Problem Solving Lab	0	0	4	2
8	EC	ASH210	Workshop/Manufacturing Practices	0	0	6	3
Total Credits							**23**

ASH202 – Mathematics-II
(Probability and Statistics)

Course Information	
Course Code	ASH202
Course Name	Mathematics-II
Category	Basic Science Courses
Pre-requisites	Basic knowledge of +2 level probability and statistics
L–T–P–C	3–1–0–4

Course Objectives

1. To make the students familiar with the basics of probability theory.
2. To explain the use of continuous and discrete probability distributions in all branches of engineering.
3. To develop the tools of basic statistics, applied statistics in connection with engineering purpose.

Course Outcomes

At the end of the course the students will be able to learn the basics of Probability and Statistics and apply them to solve engineering problems.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Basic Probability** Probability spaces, conditional probability, independence; theorem of total probability and Bayes' theorem.
UNIT II	**Random Variables** Discrete random variables, continuous random variables, independent random variables, mean and variance of random variables, distribution functions and densities.
UNIT III	**Discrete Probability Distributions** Binomial and Poisson distributions, Poisson approximation to the binomial distribution, infinite sequences of Bernoulli trials; Correlation coefficient.
UNIT IV	**Continuous Probability Distributions** Normal, exponential and gamma distributions with their properties.

Unit	Course Contents
UNIT V	**Basic and Applied Statistics** Measures of Central tendency, Test of significance. Curve fitting by the method of least squares—fitting of straight lines, second degree parabolas and more general curves.

Text Books

Text Books
1. E. Kreyszig, *Advanced Engineering Mathematics*, 9th Edition, John Wiley & Sons, 2006. (Unit-I–V)
2. S. Ross, *A First Course in Probability*, 6th Ed., Pearson Education India, 2002. (Unit-I–V)

References

References
1. W. Feller, *An Introduction to Probability Theory and its Applications*, Vol. 1, 3rd Ed., Wiley, 1968.
2. N. P. Bali and M. Goyal, *A Text Book of Engineering Mathematics*, Laxmi Publications, Reprint, 2010.

ASH203 – Chemistry-I

Course Information	
Course Code	ASH203
Course Name	Chemistry-I
Category	Basic Science Course
L–T–P–C	3–1–0–4

Course Objectives

1. To understand the basic concepts of bonding and shapes of atoms, molecules and solids.
2. To understand the concepts of crystal field theory of transition metal ions.
3. To understand the basic concepts of different spectroscopic techniques and its applications in emerging fields.
4. To study the interaction forces of gases and potential energy surfaces of molecules.
5. To study the different thermodynamic functions.
6. To study the basic concepts of oxidation and reduction.
7. To study the different properties of periodic table.
8. To understand the phenomenon of isomerism and optical activity.
9. To study different types of organic reactions.
10. Chemistry lab will provide students with the first-hand experience of verifying various theoretical concepts learnt in theory courses.

Course Outcomes

1. Analyse microscopic chemistry in terms of atomic and molecular orbitals and intermolecular forces.
2. Rationalise bulk properties and processes using thermodynamic considerations.
3. Distinguish the ranges of the electromagnetic spectrum used for exciting different molecular energy levels in various spectroscopic techniques.
4. Rationalize periodic properties such as ionization potential, electronegativity, oxidation states and electronegativity.
5. Learnt the different types of organic reactions.
6. Apply basic knowledge of chemistry to solve real-world problems.
7. Estimate rate constants of reactions from concentration of reactants/products as a function of time.
8. Measure molecular/system properties such as surface tension, viscosity, conductance of solutions, redox potentials, chloride content of water, etc.
9. Synthesize a small drug molecule and analyses a salt sample.

Detailed Syllabus

Unit	Course Contents
UNIT I	Atomic and molecular structure: Schrodinger equation. Particle in a box solutions and its applications in conjugated molecules and nanoparticles. Hydrogen atom wave functions and its plots to explore their spatial variations. Equations for atomic and molecular orbitals. Molecular orbitals and energy level diagrams of diatomic molecules, butadiene and benzene. Aromaticity. Crystal field theory of transition metal ions and magnetic properties. Band structure of solids and the role of doping on band structures.
UNIT II	Spectroscopic techniques and applications: Principles of spectroscopy and selection rules: Electronic spectroscopy. Fluorescence and its applications in medicine. Vibrational and rotational spectroscopy of diatomic molecules. Nuclear magnetic resonance and magnetic resonance imaging, surface characterization techniques.
UNIT III	Intermolecular forces, potential energy surfaces and use of free energy in chemical equilibria. Ionic, dipolar and van Der Waals interactions. Equations of state of real gases and critical phenomena. Potential energy surfaces of H3, and H2F and trajectories on these surfaces. Thermodynamic functions: energy, entropy and free energy. Cell potentials, e.m.f, the Nernst equation and applications. Acid-base, oxidation-reduction. Battery technology: Rechargeable batteries. Fuel cells. Photovoltaics, Water chemistry. Corrosion. Ellingham diagrams.
UNIT IV	Periodic properties: Effective nuclear charge, variations of s, p, d and f orbital energies of atoms in the periodic table, electronic configurations, atomic and ionic sizes, ionization energies, electron affinity and electronegativity, polarizability, oxidation states, coordination numbers, hard soft acids and bases.
UNIT V	Stereochemistry, organic reactions and synthesis of drug molecule. Representations of 3 dimensional structures, Isomerism in organic molecules and transition metal compounds, optical activity. Introduction to organic reactions involving substitution, addition and elimination. Synthesis of a commonly used drug molecule.

Suggested Text Books

1. University chemistry, by B. H. Mahan
2. Chemistry: Principles and Applications, by M. J. Sienko and R. A. Plane
3. Fundamentals of Molecular Spectroscopy, by C. N. Banwell
4. Engineering Chemistry (NPTEL Web-book), by B. L. Tembe, Kamaluddin and M. S. Krishnan
5. Physical Chemistry by P. W. Atkins
6. Organic Chemistry: Structure and Function by K. P. C. Volhardt and N. E. Schore, 5th Edition

ASH204 – English

Course Information	
Course Code	ASH204
Course Name	English
Category	Humanities and Social Sciences including Management courses
L–T–P–C	2–0–2–3

Course Objectives

1. To improve the language proficiency of students in English with an emphasis on Vocabulary, Grammar, Reading and Writing skills.
2. To assist the students to carry on the tasks and activities through guided instructions and materials.
3. To integrate English language learning with employability skills.
4. To provide hands-on experience through group and individual presentations.

Course Outcomes

The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Vocabulary Building** 1.1 The concept of Word Formation 1.2 Word Families and Grammatical Categories 1.3 Prefixes and suffixes in English to form derivatives 1.4 Synonyms, Antonyms, and Homonyms.
UNIT II	**Basic Writing Skills** 2.1 Sentence Structures 2.2 Use of phrases and clauses in sentences 2.3 Creating coherence 2.4 Organizing principles of paragraphs in documents.
UNIT III	**Identifying Common Errors in Writing** 3.1 Subject-verb agreement 3.2 Noun-pronoun agreement 3.3 Misplaced modifiers 3.4 Prepositions.

Unit	Course Contents
UNIT IV	**Nature and Style of Sensible Writing** 4.1 Descriptive and Analytical Writing 4.2 Précis Writing 4.3 Writing Paragraphs 4.4 Comprehension.
UNIT V	**Oral Communication** 5.1 Listening Comprehension 5.2 Conversation in Common Everyday Situations 5.3 Communication at Workplace 5.4 Formal Presentations 5.5 Pronunciation, Intonation, Stress.

Text Books / References

Text Books / References
1. Swan, Michael. *Practical English Usage*. 2nd ed., Oxford University Press, 2016.
2. Wood, Frederick T. *A Remedial English Grammar for Foreign Students*, Trinity Press, 2022.
3. Zinsser, William. *On Writing Well: The Classic Guide to Writing Nonfiction*, Harper Collins, 2006.
4. Hamp-Lyons, Liz, and Ben Heasley. *Study Writing: A Course in Writing Skills for Academic Purposes*. 2nd ed., Cambridge University Press, 2006.
5. Kumar, Sanjay, and Pushp Lata. *Communication Skills*. 3rd ed., Oxford UP, 2024.
6. Central Institute of English and Foreign Languages. *Exercises in Spoken English*. Parts I–III, Oxford UP, 1997.

ASH206 – Chemistry-I Lab

Course Information	
Course Code	ASH206
Course Name	Chemistry-I Lab
Category	Basic Science Course
L–T–P–C	0–0–2–1

Chemistry-I Laboratory [L : 0; T : 0; P : 2 (1 credit)]
Experiments from the following:

No.	**Experiment**
1	Determination of surface tension and viscosity
2	Thin layer chromatography for separation and identification of mixtures
3	Ion exchange column for removal of hardness of water
4	Determination of chloride content of water
5	Colligative properties using freezing point depression
6	Determination of the rate constant of a reaction
7	Determination of cell constant and conductance of solutions
8	Potentiometric - determination of redox potentials and e.m.f
9	Synthesis of a polymer/drug
10	Saponification/acid value of oil
11	Chemical analysis of a salt
12	Lattice structures and packing of spheres
13	Models of potential energy surfaces
14	Chemical oscillations- Iodine clock reaction
15	Determination of the partition coefficient of a substance between two immiscible liquids
16	Adsorption of acetic acid by charcoal
17	Use of the capillary viscometers to demonstrate the isoelectric point as the pH of minimum viscosity for gelatin sols and/or coagulation of the white part of egg.

ASH207 – Programming for Problem Solving

Course Information	
Course Code	ASH207
Course Name	Programming for Problem Solving
Category	Engineering Science Courses
L–T–P–C	3–0–0–3

Course Objectives

1. To learn the fundamentals of computers.
2. To understand the various steps in program development.
3. To learn the syntax and semantics of C programming language.
4. To learn the usage of structured programming approach in solving problems.
5. To understated and formulate algorithm for programming script.
6. To analyze the output based on the given input variables.

Course Outcomes

The student will learn

1. To formulate simple algorithms for arithmetic and logical problems.
2. To translate the algorithms to programs (in C language).
3. To test and execute the programs and correct syntax and logical errors.
4. To implement conditional branching, iteration and recursion.
5. To use arrays, pointers and structures to formulate algorithms and programs.
6. To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
7. To apply programming to solve simple numerical method problems, namely root finding of function, differentiation of function and simple integration.

Detailed Syllabus

Unit	Course Contents
UNIT I	Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.). Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples. From algorithms to programs; source code, variables (with data types), variables and memory locations, Syntax and Logical Errors in compilation, object and executable code.
UNIT II	Arithmetic expressions and precedence. Conditional Branching and Loops, Writing and evaluation of conditionals and consequent branching, Iteration and loops. Arrays (1-D, 2-D), Character arrays and Strings.
UNIT III	Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required). Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference.
UNIT IV	Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort. **Structures** Defining structures and Array of Structures.
UNIT V	Idea of pointers. Defining pointers. Use of Pointers in self-referential structures. Notion of linked list (no implementation). File handling (only if time is available, otherwise should be done as part of the lab).

ASH208 – Universal Human Values

Course Information	
Course Code	ASH208
Course Name	Universal Human Values
Category	Humanities and Social Sciences including Management Courses
L–T–P–C	2–1–0–3

Course Objectives

1. To help the students appreciate the essential complementarily between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.
4. To provide a much-needed orientational input in value education to the young enquiring minds.

Course Outcomes

By the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature); they would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.

They would have better critical ability. They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society). It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

Therefore, the course and further follow up is expected to positively impact common graduate attributes like:

1. Holistic vision of life.
2. Socially responsible behaviour.
3. Environmentally responsible work.
4. Ethical human conduct.
5. Having Competence and Capabilities for Maintaining Health and Hygiene.
6. Appreciation and aspiration for excellence (merit) and gratitude for all.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Introduction to Value Education** Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education). Understanding Value Education. Self-exploration as the Process for Value Education. Continuous Happiness and Prosperity – the Basic Human Aspirations. Happiness and Prosperity – Current Scenario. Method to Fulfil the Basic Human Aspirations.
UNIT II	**Harmony in the Human Being** Understanding Human Being as the Co-existence of the Self and the Body. Distinguishing between the Needs of the Self and the Body. The Body as an Instrument of the Self. Understanding Harmony in the Self. Harmony of the Self with the Body. Programme to ensure Self-regulation and Health.
UNIT III	**Harmony in the Family and Society** Harmony in the Family – the Basic Unit of Human Interaction. 'Trust' – the Foundational Value in Relationship. 'Respect' – as the Right Evaluation. Other Feelings. Justice in Human-to-Human Relationship. Understanding Harmony in the Society. Vision for the Universal Human Order.
UNIT IV	**Harmony in the Nature/Existence** Understanding Harmony in the Nature. Interconnectedness, Self-regulation and Mutual Fulfilment among the Four Orders of Nature. Realizing Existence as Co-existence at All Levels. The Holistic Perception of Harmony in Existence.
UNIT V	**Implications of the Holistic Understanding – a Look at Professional Ethics** Natural Acceptance of Human Values. Definitiveness of (Ethical) Human Conduct. A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order. Competence in Professional Ethics. Holistic Technologies, Production Systems and Management Models – Typical Case Studies. Strategies for Transition towards Value-based Life and Profession.

Text Books
1. A Foundation Course in Human Values and Professional Ethics, R. R. Gaur, R. Asthana, G. P. Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN: 978-93-87034-47-1.
2. Human Values, A. N. Tripathi, New Age International Publishers, New Delhi, 2004.

ASH209 – Programming for Problem Solving Lab

Course Information	
Course Code	ASH209
Course Name	Programming for Problem Solving Lab
Category	Engineering Science Courses
L–T–P–C	0–0–4–2

Tutorial	**Laboratory Exercise**
Tutorial 1: Problem solving using computers	Lab 1: Familiarization with programming environment.
Tutorial 2: Variable types and type conversions	Lab 2: Simple computational problems using arithmetic expressions.
Tutorial 3: Branching and logical expressions	Lab 3: Problems involving if-then-else structures.
Tutorial 4: Loops, while and for loops	Lab 4: Iterative problems, e.g., sum of series.
Tutorial 5: 1D Arrays: searching, sorting	Lab 5: 1D Array manipulation.
Tutorial 6: 2D Arrays and Strings	Lab 6: Matrix problems, String operations.
Tutorial 7: Functions, call by value	Lab 7: Simple functions.
Tutorial 8 & 9: Numerical methods (Root finding, numerical differentiation, numerical integration)	Lab 8 and 9: Programming for solving Numerical methods problems.
Tutorial 10: Recursion, structure of recursive calls	Lab 10: Recursive functions.
Tutorial 11: Pointers, structures and dynamic memory allocation	Lab 11: Pointers and structures.
Tutorial 12: File handling	Lab 12: File operations.

ASH210 – Workshop / Manufacturing Practices

Course Information	
Course Code	ASH210
Course Name	Workshop / Manufacturing Practices
Category	Engineering Science Courses
L–T–P–C	1–0–4–3

Course Objectives

1. To provide exposure to the students with hands-on experience on various basic engineering practices in Civil, Mechanical, Electrical and Electronics Engineering.
2. To have a study and hands-on exercise on plumbing and carpentry components.
3. To have a practice on gas welding, foundry operations and fitting.
4. To have a study on measurement of electrical quantities, energy and resistance to earth.
5. To have a practice on soldering.

Course Outcomes

Upon completion of this course, the students will gain knowledge of the different manufacturing processes which are commonly employed in the industry, to fabricate components using different materials.

Upon completion of this laboratory course, students will be able:

1. To fabricate components with their own hands.
2. To relate practical knowledge of the dimensional accuracies and dimensional tolerances possible with different manufacturing processes.
3. To design small devices of their interest by assembling different components.

Detailed Syllabus

Module	Course Contents
Module I	Manufacturing Methods – casting, forming, machining, joining, advanced manufacturing methods.
Module II	CNC machining, Additive manufacturing.
Module III	Fitting operations & power tools.
Module IV	Carpentry.
Module V	Plastic moulding, glass cutting.
Module VI	Metal casting.
Module VII	Welding (arc welding & gas welding), brazing.

Practicals

No.	Practical
1	Machine shop.
2	Fitting shop.
3	Carpentry.
4	Welding shop (Arc welding + Gas welding).
5	Casting.
6	Smithy.
7	Plastic moulding & Glass Cutting.

Text Books

Text Books
1. Hajra Choudhury S.K., Hajra Choudhury A.K. and Nirjhar Roy S.K., *Elements of Workshop Technology*, Vol. I (2008) and Vol. II (2010), Media Promoters and Publishers Private Limited, Mumbai.

References

References
1. Kalpakjian S. and Steven S. Schmid, *Manufacturing Engineering and Technology*, 4th Edition, Pearson Education India Edition, 2002.
2. Gowri P. Hariharan and A. Suresh Babu, *Manufacturing Technology – I*, Pearson Education, 2008.
3. Roy A. Lindberg, *Processes and Materials of Manufacture*, 4th Edition, Prentice Hall India, 1998.
4. Rao P.N., *Manufacturing Technology*, Vol. I and Vol. II, Tata McGraw Hill House, 2017.

DETAILED SYLLABUS
Second Year – 3rd Semester

Sl.	Type	Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE301	Discrete Mathematics & Graph Theory	3	1	0	4
2	PC	CSE302	Data Structures	3	0	0	3
3	EC	CSE303	Digital Electronics	3	0	0	3
4	PC	CSE304	Object Oriented Programming	3	0	0	3
5	PC	CSE305	Data Structures Lab	0	0	4	2
6	EC	CSE306	Digital Electronics Lab	0	0	4	2
7	PC	CSE307	Object Oriented Programming Lab	0	0	4	2
8	BC	ASH301	Mathematics–III	2	0	0	2
9	HC	ASH302	Effective Technical Communication	3	0	0	3
Total Credits							**24**

CSE301 – Discrete Mathematics and Graph Theory

Course Information	
Course Code	CSE301
Course Name	Discrete Mathematics and Graph Theory
Category	Professional Core Course (PC)
Credits	L:3 T:1 P:0 C:4
Pre-Requisites	Elementary algebra and arithmetic

Course Objectives

Throughout the course, students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following:

1. Use mathematically correct terminology and notation.
2. Apply logical reasoning to solve a variety of problems.
3. Understand discrete mathematical structures.
4. Formulate and solve graph problems.

Course Outcomes

1. For a given logic sentence, express it in terms of predicates, quantifiers, and logical connectives.
2. For a given problem, derive the solution using deductive logic and prove the solution based on logical inference.
3. For a given mathematical problem, classify its algebraic structure.
4. Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.
5. Develop the given problem as a graph network and solve it with techniques of graph theory.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Sets and Relations** Set Basics, Basic operation on sets, Cartesian product, Venn Diagram, counting principles, Inclusion and Exclusion principle, pigeon-hole principle, Mathematical Induction. Relations: Relation Basics, Types and properties of relations, Inductive form of relations, Congruence relations on semigroups, equivalence relations and partitions, Partially Ordered Set, Lattice.

Unit	Course Contents
UNIT II	**Functions and Algebraic Structures** Function types, mapping in functions, composition of functions, inverse functions, commutative diagrams, special functions, monotone functions. Algebraic Structures: Definition and properties of groups, semigroups, monoids, rings, fields, vector spaces, and Boolean Algebra.
UNIT III	**Recursion and Recurrence Relation; Propositional Logic** Recursion and Recurrence Relation: Basic idea, Sequence and discrete function. Generating functions and applications. Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. Proof Techniques: Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency.
UNIT IV	**Introduction to Graph Theory and Trees** Graphs, Digraphs, Isomorphism, Walks, Paths, Circuits, Shortest Path Problem, Dijkstra's Algorithm, Trees, Properties of Trees, Cotrees and Fundamental Circuits.
UNIT V	**Graph Theoretic Algorithms and Applications** Shortest Spanning Trees – Kruskal's Algorithm, Prim's Algorithm, DFS, BFS, Cut Sets, Fundamental Cut Sets and Cut Vertices, Planar and Dual Graphs, Graph Colouring, Metric Representation of Graphs, Networks, Flow Augmenting Path, Ford–Fulkerson Algorithm for Maximum Flow.

Text Books

1. Kolman, Busby and Ross, *Discrete Mathematical Structures*, 6th Edition, PHI, 2009.
2. Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, 7th Edition, McGraw Hill, 2011.
3. Deo N., *Graph Theory with Applications to Engineering and Computer Science*, PHI, 2004.
4. C. L. Liu and D. P. Mohapatra, *Elements of Discrete Mathematics: A Computer Oriented Approach*, 3rd Edition, Tata McGraw–Hill.

Reference Books

1. R. Balakrishnan and K. Ranganathan, *A Text Book of Graph Theory*, 2nd Edition, MH, 2013.
2. Tremblay and Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw Hill, 2001.
3. Tremblay and Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw Hill, 2001.

CSE302 – Data Structures

Course Information	
Course Code	CSE302
Course Name	Data Structures
Category	Professional Core Course (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Any computer language preferably C (Desirable)

Course Objectives

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques.
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures.

Course Outcomes

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.
2. For a given Search problem (Linear Search and Binary Search) student will able to implement it.
3. For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.
4. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.
5. Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Fundamentals** Basic Terminologies: Elementary Data Organizations. Time and Space analysis of Algorithms: Time Complexity, Space Complexity, Order Notations. Recursion – Design of recursive algorithms. Searching: Linear Search and Binary Search Techniques and their complexity analysis.
UNIT II	**Stacks and Queues** ADT Stack and its operations: Algorithms and their complexity analysis. Applications of Stacks: Expression Conversion and Evaluation – corresponding algorithms and complexity analysis. ADT Queue. Types of Queue: Simple Queue, Circular Queue, Priority Queue. Operations on each type of Queue: Algorithms and their analysis.
UNIT III	**Linked Lists** Singly Linked Lists: Representation in memory. Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list. Linked representation of Stack and Queue. Header nodes. Doubly Linked List: Operations on it and algorithmic analysis. Circular Linked Lists: All operations, their algorithms and the complexity analysis.
UNIT IV	**Trees** Basic Tree Terminologies. Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree. Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: Definitions, algorithms and analysis.
UNIT V	**Sorting and Hashing** Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort. Performance and comparison among all the methods. Hashing. **Graph** Basic Terminologies and Representations. Graph search and traversal algorithms and complexity analysis.

Text Books

Text Books
1. S. K. Srivastava and Deepali Srivastava, *Data Structure through C in Depth*, BPB Publications, 2004.
2. Ellis Horowitz, Sartaj Sahni and S. A. Freed, *Fundamentals of Data Structures in C* (Second Edition), Universities Press, 2008.

Reference Books

Reference Books
1. Mark Allen Weiss, *Algorithms, Data Structures, and Problem Solving with C++*, Illustrated Edition, Pearson, 1995.
2. R. G. Dromey, *How to Solve It by Computer*, 2nd Impression, Pearson Education.

CSE303 – Digital Electronics

Course Information	
Course Code	CSE303
Course Name	Digital Electronics
Category	Engineering Science Course (ESC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Fundamental knowledge of electronics and electrical circuits.

Course Objectives

1. Introduce students to the Digital Systems, learn about number systems, Boolean algebra and logic gates.
2. Students learn about the representation, manipulation, and minimization of Boolean functions.
3. Students should be able to learn how to design combinational and sequential circuits.
4. Students should be able to understand the concept of finite state machines, state minimization, and algorithmic state machines.
5. Learn about analysis and synthesis of asynchronous circuits.

Course Outcomes

1. Students will be able to explain the concept of digital systems, number systems which helps digital representation of information.
2. Students will be able to explain the basic logic operation of NOT, AND, OR, NAND, NOR, X-OR and X-NOR.
3. Students will be capable of understanding the different type of logic families like DTL, TTL, ECL, MOS, CMOS, etc., and their operation, design, and specification.
4. Students will be able to interpret the Boolean algebra expressions, logic functions, circuits, and truth tables. Also, learn the minimization techniques of Boolean algebra expressions.
5. Students will be able to design the combinational circuits and analyze the computer software application. Also, learn the detail concept and synthesis approaches of threshold logic.
6. Students will be able to understand the detail concept of memory elements like latches and edge-triggered flip-flops.
7. Students will be able to design the synchronous sequential circuit, and also able to implement the computer software application.
8. Students will be able to understand the concept of registers, counters and their applications in digital circuits. Moreover, students will gain knowledge of the detail designing procedure of asynchronous sequential circuits.
9. Students will be able to model and analyze the A/D and D/A conversion technique. Also, able to understand the different types of memory devices.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Fundamentals of Digital Systems and Number Systems** Introduction: Need of Digital Systems, Digital Vs Analog Systems, Logic Levels and Pulse Waveforms, Elements & Functions of Digital Logic, and Benefits of Digital Systems. Number Systems: Systematic way to represent and manipulate number systems, Signed and Unsigned number representation, Binary arithmetic, BCD, Gray-Code, XS-3 code representation, Error Detection and Correction code.
UNIT II	**Logic Gates, Logic Families to Implement Gates, Boolean Algebra** Logic Gates: Binary Logic, Importance of Moore's Law, Introduction of Logic Gates. Logic Families: DTL, TTL, ECL, MOS, CMOS etc. and their operation, design, and specifications. Boolean Algebra or Switching Algebra: Basic concept of Boolean Algebra, Basic Laws and Properties of Boolean Algebra, Definition of Boolean Functions and their properties, Boolean Function representation, manipulation and minimization (by algebraic method, Karnaugh Map method, Quine-McCLUSKY method).
UNIT III	**Combinational Logic and Threshold Logic Design** Combinational Logic Design: Introduction of combinational circuits, and design procedure of combinational circuit modules, Binary Adder-Subtractor, Decimal Adder, Comparator, Decoder, Encoder, Multiplexer, De-Multiplexer, Parity Generator. Threshold Logic Design: Basic concept of threshold logic and importance, Threshold element and construction of threshold gate, Boolean function realization using threshold gate, Synthesis of threshold function.
UNIT IV	**Sequential Logic Design** Introduction: Basic concept of memory elements like Latches and Flip-Flops, Design of Latches, Notion of Clock, Design of Flip-Flops, Clocking and Timing. Synthesis of Synchronous Sequential Circuits: Combinational Vs Sequential Circuits, Finite State Machine (FSM), Model of Synchronous Sequential Machine, State Transition Diagram and State Table, Examples of Synchronous Sequential Circuits design methodology. Design of Registers and Counters: Different variations of Registers and their design, Design of asynchronous and synchronous counters. Design of Asynchronous Sequential Circuits.
UNIT V	**Analog-to-Digital (A/D), Digital-to-Analog (D/A) Conversion, Memory Devices** A/D, D/A Conversion: Basic concept of D/A Conversion, Different types of D/A converters and conversion techniques. Memory Devices: RAM, ROM, EPROM, EEPROM.

Text Books

Text Books

1. M. Morris Mano and Michael D. Ciletti, *Digital Design*, 4th Edition, Pearson Education, 2007.
2. A. Anand Kumar, *Fundamental of Digital Circuits*, 4th Edition, PHI Learning Private Limited, 2016.

Reference Books

Reference Books
1. R. P. Jain, *Modern Digital Electronics*, 4th Edition, TMH, 2010.
2. Zvi Kohavi and Niraj K. Jha, *Switching and Finite Automata Theory*, 3rd Edition, Cambridge University Press, 2010.

CSE304 – Object Oriented Programming

Course Information	
Course Code	CSE304
Course Name	Object Oriented Programming
Category	Professional Core Course (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Programming for Problem Solving, Data Structures (Desirable)

Course Objectives

1. To introduce students to the Object-oriented Programming paradigm and its core principles.
2. To build proficiency in Java syntax, program structure, and the Java execution model (JVM/JRE/JDK).
3. To develop the ability to design class hierarchies using inheritance, interfaces, and polymorphism.
4. To familiarize students with exception handling, multithreading, and the Java Collections Framework.
5. To introduce modern Java programming constructs – lambda expressions, functional interfaces, and the Stream API.

Course Outcomes

On completion of this course, students will be able to:

1. Specify simple abstract data types and design their implementations using abstraction functions.
2. Apply object-oriented design principles such as encapsulation, inheritance, polymorphism and interface-based design to model real-world problems.
3. Identify and apply common object-oriented design patterns with appropriate examples.
4. Handle runtime exceptions and multithreaded execution safely and effectively in Java programs.
5. Use the Java Collections Framework and functional programming constructs (lambdas, streams) to write concise, efficient code.
6. Design and implement a simple event-driven graphical user interface application in Java.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Introduction to Java and OOP Concepts** Need for OOP, history and evolution of Java, core OOP concepts (encapsulation, abstraction, inheritance, polymorphism) and their benefits. Overview and characteristics of Java; JDK, JRE and JVM; the Java program compilation and execution process; bytecode; the JVM as an interpreter and just-in-time compiler. Setting up a Java development environment; writing, compiling and running a simple Java program.
UNIT II	**Data Abstraction and Inheritance** Data types and literals, variables, arrays, operators, control flow statements. Classes and objects, constructors, access and class member modifiers, the `this` keyword, static members. Inheritance basics, member access, method overriding vs. overloading, creating multilevel hierarchies, use of `super`, using `final` with inheritance. Interfaces – defining, implementing, nested interfaces, default and static methods in interfaces, variables in interfaces, extending interfaces.
UNIT III	**Polymorphism, Exceptions and Packages** Polymorphism – ad hoc and pure polymorphism, method overriding, abstract classes, the Object class, anonymous inner classes, wrapper classes, autoboxing/unboxing. Exception handling – try, throw, catch, finally, throwing and catching exceptions, re-throwing exceptions, custom exceptions, exception specifications, try-with-resources. Packages – defining a package, CLASSPATH, access protection, importing packages; introduction to Javadoc documentation.
UNIT IV	**Multithreading, Collections and Functional Programming** Thread life cycle, creating threads, thread priority, synchronization, blocked states, extending the Thread class, the Runnable interface, starting threads, basics of concurrency utilities. Collections overview, the Collection interfaces, Collection classes – ArrayList, LinkedList, HashSet, TreeSet, PriorityQueue, ArrayDeque, HashMap/TreeMap, Comparable and Comparator. Introduction to functional programming in Java – lambda expressions, functional interfaces (Runnable, Comparator, Predicate, Function), method references, and an overview of the Stream API for filtering, mapping and reducing collections.
UNIT V	**Files, I/O Streams and GUI Programming** Stream classes – byte streams and character streams, reading console input and writing console output, the File class, reading and writing files, buffered and object streams, serialization basics. Introduction to GUI programming with JavaFx – stage and scene, layouts, controls, event handling; overview of Swing as a legacy alternative. Mini-project component: a small event-driven desktop application applying OOP design principles covered in the course.

Text Books

Text Books
1. Herbert Schildt, *Java: The Complete Reference*, McGraw Hill.
2. E. Balagurusamy, *Programming with Java: A Primer*, McGraw Hill.
3. Cay S. Horstmann, *Core Java Volume I – Fundamentals*, Pearson.

Reference Books

Reference Books
1. R. K. Das, *Core Java for Beginners*, Vikas Publishing.
2. Paul Deitel and Harvey Deitel, *Java: How to Program*, Pearson.
3. Joshua Bloch, *Effective Java*, Addison-Wesley.
4. Kathy Sierra and Bert Bates, *Head First Java: A Brain-Friendly Guide*, 2nd Edition.

CSE305 – Data Structures Lab

Course Information	
Course Code	CSE305
Course Name	Data Structures Lab
Category	Professional Core Course (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	NIL

Course Objectives

1. Identify, formulate, review research literature, and analyze complex engineering problems.
2. Design solutions for complex engineering problems and design system components or processes that meet the specified needs.
3. Create, select, and apply appropriate data structures for different problems.

Laboratory Experiments

No.	Experiment
1	Create a dynamic array.
2	Implement pointer operations.
3	Implement binary search.
4	Create singly linked list and doubly linked list and perform: (a) Insertion (b) Deletion (c) Display
5	Implement stack using array and linked list.
6	Implement queue and circular queue using array and linked list.
7	Perform the following operations for stack and queue: (a) Insertion (b) Deletion (c) Display
8	Write a C program that uses stack operations: (a) To convert a given infix expression into its postfix equivalent. (b) Evaluate postfix expression. (c) Check for balanced parenthesis.
9	Implement the following: (a) Binary Search Tree and its traversal. (b) Graph traversal algorithms.

No.	Experiment
10	Implement the following sorting algorithms: (a) Insertion Sort (b) Selection Sort (c) Bubble Sort (d) Merge Sort (e) Quick Sort

Course Outcomes

1. Demonstrate knowledge and understanding of the problem and the nature of solution.
2. Gain hands-on experience and apply the principles of data structures.
3. Apply reasoning informed by the appropriate knowledge to assess different problem.

CSE306 – Digital Electronics Lab

Course Information	
Course Code	CSE306
Course Name	Digital Electronics Lab
Category	Engineering Science Course (ESC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	Fundamental knowledge of electronics and basic C programming skill.

Course Objectives

1. Introduce students to the Digital Systems, learn about number systems, Boolean algebra and logic gates.
2. Students learn about the representation, manipulation, and minimization of Boolean functions.
3. Students should be able to learn how to design combinational and sequential circuits through Hardware programming design using Verilog/VHDL.
4. Students should be able to understand the detail circuit structure and their behaviours with the help of software application.
5. Students should be able to learn about the details behaviour of combinational and sequential circuit through Hardware Programming Language.

Laboratory Experiments

No.	Experiment
1	Introduction to Verilog/VHDL language.
2	Design of all basic and Universal gates using Verilog/VHDL.
3	Design of XOR and XNOR gate using VHDL.
4	Design of Full Adder and Full Subtractor using VHDL.
5	Design of 4-bit Parallel Adder-Subtractor using VHDL.
6	Design of 4:1 Multiplexer using VHDL.
7	Design of 1:4 De-multiplexer using VHDL.
8	Design of 8×3 Encoder using VHDL.
9	Design of 3×8 Decoder using VHDL.
10	Design of Priority Encoder using VHDL.
11	Design of 4-bit Array Multiplier using VHDL.
12	Design of S-R Flip-Flop using VHDL.
13	Design of D Flip-Flop using VHDL.
14	Design of J-K/T Flip-Flop using VHDL.
15	Design of Master-Slave Flip-Flop.

No.	Experiment
16	Design of Mod-10 Synchronous Counter.
17	Design Ripple Counter / Twisted Ring Counter.

Text Books

Text Books
1. M. Morris Mano and Michael D. Ciletti, *Digital Design*, 4th Edition, Pearson Education, 2007.
2. A. Anand Kumar, *Fundamental of Digital Circuits*, 4th Edition, PHI Learning Private Limited, 2016.
3. Douglas L. Perry, *VHDL Programming by Example*, 4th Edition, McGraw-Hill, 2012.

Course Outcomes
1. Students will be able to design the combinational and sequential circuit, and also able to implement the computer software application. 2. Students will be able to explain the details behaviour of the varieties of digital circuits.

CSE307 – Object Oriented Programming Lab

Course Information	
Course Code	CSE307
Course Name	Object Oriented Programming Lab
Category	Professional Core Course (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	CSE304 – Object Oriented Programming

Course Objectives

1. To provide hands-on experience with the Java programming environment and standard development tools.
2. To enable students to implement and apply core OOP principles – encapsulation, inheritance, and polymorphism – through practical exercises.
3. To build proficiency in exception handling, multithreading, and use of the Java Collections Framework in working programs.
4. To give practical exposure to modern Java constructs – lambda expressions and the Stream API.
5. To enable students to design and build a small event-driven GUI application using JavaFX.

Laboratory Experiments

No.	Experiment
1	Install and configure a Java development environment (JDK, IDE); write, compile and execute a basic Java program; explore JVM bytecode using `javap`.
2	Programs using variables, arrays, operators and control flow statements to solve simple computational problems.
3	Programs demonstrating class design – defining classes, objects, constructors, and use of the `this` keyword and static members.
4	Programs demonstrating inheritance – single and multilevel hierarchies, method overriding, and use of `super` and `final`.
5	Programs demonstrating interfaces – defining and implementing interfaces, use of default/static interface methods, and multiple interface implementation.
6	Programs demonstrating polymorphism – method overloading, dynamic method dispatch, abstract classes, and use of the `Object` class.
7	Programs demonstrating exception handling – `try`/`catch`/`finally`, multiple catch blocks, custom exception classes, and `try`-with-resources.
8	Programs demonstrating package creation and use, including access protection across packages.
9	Programs demonstrating multithreading – creating threads via `Thread` and `Runnable`, thread priorities, and basic synchronization.

No.	Experiment
10	Programs using the Java Collections Framework – `ArrayList`, `LinkedList`, `HashSet`, `TreeSet`, `HashMap`, and custom sorting with `Comparable`/`Comparator`.
11	Programs demonstrating lambda expressions, functional interfaces, method references, and Stream API operations (`filter`, `map`, `reduce`) on collections.
12	Programs demonstrating file handling – reading/writing text files using byte and character streams, and console I/O.
13	Mini-project: Design and implement a small event-driven GUI application in JavaFX (e.g., a simple calculator, to-do list, or student record manager) applying the OOP principles covered in the course.

Course Outcomes

On completion of this lab, students will be able to:

1. Set up and use a Java development environment to write, compile, and debug programs.
2. Apply core OOP principles – encapsulation, inheritance, polymorphism, and interfaces – to implement working solutions.
3. Write robust programs using exception handling and basic multithreading.
4. Use the Java Collections Framework and Stream API to process data efficiently.
5. Design and implement a simple event-driven graphical user interface application.

ASH301 – Mathematics–III
(Calculus, Differential Equations and Algebraic Structures)

Course Information	
Course Code	ASH301
Course Name	Mathematics–III
Category	Basic Science Course
Credits	L:2 T:0 P:0 C:2
Pre-Requisites	Basic knowledge of single variable calculus and differential equations.

Course Objectives

1. To present the effective mathematical tools for the solutions of differential equations that model physical processes.
2. To introduce the tools of different multivariable calculus and algebraic structures that are used in the modeling of various engineering problems.

Course Outcomes

At the end of the course the students should be able to

1. Understand the basics of ordinary differential equations and their engineering applications.
2. Get familiar with the concept of sequences, series, multivariable calculus and algebraic structures and their applications.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Sequences and Series** Convergence of sequence and series, tests for convergence; Power series, Taylor's series.
UNIT II	**Multivariable Calculus** Limit, continuity and partial derivatives, directional derivatives, total derivative.
UNIT III	**First Order Ordinary Differential Equations** Exact, linear and Bernoulli's equations, Euler's equations.
UNIT IV	**Ordinary Differential Equations of Higher Degree and Higher Orders** Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.

Unit	Course Contents
UNIT V	**Algebraic Structures** Algebraic structures with one binary operation – semigroup, monoid and group. Cosets, Lagrange's theorem, normal subgroup.

Text Books

Text Books
1. G. B. Thomas and R. L. Finney, *Calculus and Analytic Geometry*, 9th Edition, Pearson, Reprint, 2002. (Unit I, II)
2. S. L. Ross, *Differential Equations*, 3rd Edition, Wiley India, 1984. (Unit III, IV)
3. J. P. Tremblay and R. P. Manohar, *Discrete Mathematics with Applications to Computer Science*, Tata McGraw-Hill, 1997. (Unit V)

Reference Books

Reference Books
1. E. A. Coddington, *An Introduction to Ordinary Differential Equations*, Prentice Hall India, 1995.
2. E. L. Ince, *Ordinary Differential Equations*, Dover Publications, 1958.
3. C. L. Liu, *Elements of Discrete Mathematics*, 2nd Edition, Tata McGraw-Hill, 2000.

ASH302 – Effective Technical Communication

Course Information	
Course Code	ASH302
Course Name	Effective Technical Communication
Category	Humanities and Social Sciences including Management Courses
Credits	L:2 T:0 P:2 C:3

Course Objectives

Considering the significance of English language as a tool for academic and professional communication, the course aims to develop and enhance the linguistic and communicative competence of the learners. This course aims to train the B.Tech students in basic principles of communication and language by developing their LSRW skills, namely listening, speaking, reading and writing skills, and thereby improving their proficiency in oral and written communication in English. During the course, the learners will be exposed to various forms of professional communication.

Course Outcomes

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

1. The barriers, challenges, and frameworks of communication.
2. Methods for speaking and presenting with confidence.
3. How to share views in a professional setting.
4. Techniques for writing professional business documents.
5. How to interact in one-on-one or group meetings.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Technical Writing Fundamentals** Characteristics of technical writing; Differences between technical and literary style; Different kinds of technical documents; Structural organization; Strategies for organization; Creating coherence and sequencing of sentences.
UNIT II	**Technical Writing Style and Language** Technical writing style and language; Punctuation; Editing strategies to achieve appropriate technical style; Study of advanced grammar; Use of appropriate diction; Avoiding redundancy.
UNIT III	**Oral Communication** Public speaking; Tempo of speech and Phrasal pause; Presentation aids; Impromptu Speech; Oral presentation; Group discussion; Interviews; Curriculum Vitae and Resume.

Unit	Course Contents
UNIT IV	**Professional Writing** Technical articles; Minutes of meetings; Event report; Project report; Official letter; Paragraph writing; Instructions; Description and Analysis.
UNIT V	**Professional Communication Etiquette** Writing etiquettes in social and office settings; Email etiquettes; Requests, Offers, Refusal, Acceptance; Initiating and closing conversations.

Text Books / References

Text Books / References
1. Beer, David F., and David A. McMurrey. *A Guide to Writing as an Engineer*. 5th ed., Wiley, 2019.
2. Jungk, Dale. *Applied Writing for Technicians*. McGraw-Hill Higher Education, 2005.
3. Raman, Meenakshi, and Sangeeta Sharma. *Technical Communication: Principles and Practice*. Fourth Edition, Oxford University Press, 2022.
4. Bovee, Courtland L., John V. Thill, and Roshan Lal Raina. *Business Communication Today*. 15th ed., Pearson India, 2021.

DETAILED SYLLABUS
Second Year – 4th Semester

Sl.	Type	Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE401	Formal Languages and Automata Theory	3	1	0	4
2	PC	CSE402	Data Science Using Python	3	0	0	3
3	PC	CSE403	Database Management Systems	3	0	0	3
4	PC	CSE404	Computer Organization and Architecture	3	1	0	4
5	PC	CSE405	Data Science Using Python Lab	0	0	4	2
6	PC	CSE406	Database Management Systems Lab	0	0	4	2
7	PC	CSE407	IT Workshop I (Web Development)	0	0	4	2
8	HC	ASH401	Management-I (Organisational Behaviour)	3	0	0	3
9	MC (Audit)	ASH402	Environmental Science	2	0	0	0
						Total Credits	**23**

Note: Students are required to undertake a mandatory **Summer Training Programme** of at least 4 weeks (40 hours) in duration at reputed industries, organizations, or research institutes after the completion of the 4th semester.

CSE401 – Formal Languages and Automata Theory

Course Information	
Course Code	CSE401
Course Name	Formal Languages and Automata Theory
Category	Professional Core Course (PC)
Credits	L:3 T:1 P:0 C:4
Pre-Requisites	Calculus, Data Structures and Algorithms

Course Objectives

1. To understand problem classification and problem solving by machines.
2. To study computing machines by describing, classifying and comparing different types of computational models.
3. Understand various Computing models like Finite State Machine, Pushdown Automata, and Turing Machine.
4. Be aware of Decidability and Un-decidability of various problems.
5. Learn types of grammars.

Course Outcomes

At the end of the course, the student should be able to:

1. Design Finite State Machine, Pushdown Automata, and Turing Machine.
2. Write mathematical expressions for the formal languages.
3. Explain the Decidability or Undecidability of various problems.
4. Apply the basic knowledge of Theory of Computation to solve computational problems and in the field of compiler design.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Introduction to Finite Automata** Introduction: Alphabet, languages and grammars. Production rules and derivation of languages. Chomsky hierarchy of languages. Regular expressions, regular languages, applications, Regular grammars, Finite Automata– DFA and NFA, conversion of NFA to DFA, NFA with null move, conversion of NFA with Null move to DFA without Null move, Equivalence of DFA and NFA, Finite Automata with output–Mealy Machine and Moore Machine, Conversion from one machine to another.

Unit	Course Contents
UNIT II	**Regular Expressions and Languages** Basics of Regular expressions, Basic Operations on RE–Kleene's theorem, Identities of RE, The Arden's theorem, Construction of Finite Automata from RE, NFA to DFA conversion using ε-Closure method, Construction of Regular Grammar from RE, Construction of FA from Regular Grammar, Closure and decision properties of regular sets. Pumping lemma of regular sets. Minimization of finite automata.
UNIT III	**Context-Free Grammars and Languages** Definitions of Context Free Grammar–Backus Naur Form (BNF), Derivation and Parse Tree, Applications of Context Free Grammars, Ambiguity in CFG. Simplification of CFG–Removal of Useless Symbols, Unit Production and Null Production. Left and Right Linear Grammars, Equivalence of Left and Right Linear Grammars. Normal Forms–Chomsky and Greibach Normal Forms, Closure properties of Context Free Languages. Pumping lemma, Ogden's lemma.
UNIT IV	**Push Down Automata and Turing Machines** Introduction to Push Down Automata, Acceptance by a PDA, Deterministic Push Down Automata and Non-deterministic Push Down Automata. Two-stack PDA, Construction of PDA from CFG and Construction of CFG equivalent to PDA. Turing Machines–Transitional representation, Conversion of RE to TM, Two-stack PDA and TM, Variations of the Turing Machine model, Turing computability, Type-0 languages, Church-Turing hypothesis, TM languages, Unrestricted grammar.
UNIT V	**Unsolvable Problems and Computational Complexity** Recursive and recursively enumerable sets and their properties, Universal languages, Reducibility and Undecidable problems, Rice's Theorem, Post Correspondence Problem (PCP) and Modified PCP. Types of computational complexity–Time and Space complexity, The Classes P and NP, P = NP? – The Million Dollar Question, NP-Complete and NP-Hard problems.

Text Books

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages and Computation*, 3rd Edition, Pearson Education, 2008.
2. John C. Martin, *Introduction to Languages and the Theory of Computation*, Fourth Edition, Tata McGraw Hill Publishing Company, 2011.

Reference Books

1. Poonam Sinha, Sunita and A. Saxena, *Theory of Computation*, Laxmi Publication, 2014.
2. H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, 2nd Edition, Prentice Hall, Englewood Cliffs, 2005.

CSE402 – Data Science Using Python

Course Information	
Course Code	CSE402
Course Name	Data Science Using Python
Category	Professional Core Course (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Programming for Problem Solving, Mathematics for Computing (Linear Algebra & Calculus)

Course Objectives

The main objectives of this course are to:

1. Balance fundamental theoretical concepts of Data Science with hands-on Python program execution.
2. Master linear algebra and statistical foundations alongside numerical computing in NumPy and SciPy.
3. Pair exploratory data analysis and visual encoding theory with practical Pandas and Seaborn workflows.
4. Derive machine learning algorithms theoretically and implement them using Scikit-Learn pipelines.
5. Formulate domain-specific data science solutions and deploy models into Python execution pipelines.

Course Outcomes

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

1. Explain core Data Science lifecycles and build optimized Python environments for analytical tasks.
2. Apply linear algebra and statistical math using vectorized operations in NumPy and SciPy.
3. Perform exploratory data analysis and generate informative plots using Pandas, Matplotlib, and Seaborn.
4. Mathematically formulate supervised and unsupervised algorithms and implement them in Scikit-Learn.
5. Execute feature engineering, evaluate model performance metrics, and serialize Python models for deployment.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Data Science Lifecycle & Python Programming Foundations** **D**efinition and scope of Data Science; Data Science Lifecycle (Problem Formulation, Data Acquisition, Cleaning, Modeling, Deployment); Data Taxonomy (Structured, Semi-structured, Unstructured); Data privacy principles and ethics. **E**nvironment setup (Jupyter, Colab, Anaconda); Python object-oriented programming for data pipelines; Advanced data structures (Lists, Dicts, Sets, Tuples); Functional constructs (Lambda, Map, Filter, Reduce, List/Dict Comprehensions); File I/O for structured formats (CSV, JSON, XML).
UNIT II	**Mathematical Computing & Numerical Methods** **L**inear Algebra for Data Science (Vectors, Matrices, Dot Products, Matrix Multiplication, Determinants, Eigenvalues & Eigenvectors); Basic Probability theory, Random Variables, and Probability Distributions (Gaussian, Binomial, Poisson). **N**umPy `ndarray` architecture; Vectorization and Broadcasting mechanics; Matrix operations and linear algebra modules (`numpy.linalg`); SciPy statistical module (`scipy.stats`); Random sampling and pseudo-random generators.
UNIT III	**Data Wrangling Theory & Visual Analytics** **P**rinciples of Data Wrangling and Normalization; Descriptive Statistics (Central Tendency, Dispersion, Skewness, Kurtosis); Correlation vs. Causation; Principles of Visual Encoding, Color Theory, and Chart Selection Rules. **P**andas Series and DataFrame operations; Data Cleaning (Missing value imputation strategies, Outlier detection/removal, String manipulation); GroupBy aggregation and Pivot tables; Visualization with Matplotlib (Pyplot/Object-Oriented API) and Seaborn (Statistical plots, Heatmaps, Pair plots).
UNIT IV	**Statistical Inference & Supervised Learning Theory** **H**ypothesis Testing (p-value, Null/Alternative Hypothesis, Type I/II Errors, t-test, ANOVA); Mathematical derivations of Linear Regression (Ordinary Least Squares), Logistic Regression (Logit function), Decision Trees (Entropy, Gini Index), and K-Nearest Neighbors. **H**ypothesis testing with SciPy; Model building using Scikit-Learn; Data Preprocessing (`StandardScaler`, `OneHotEncoder`); Train-Test Split and K-Fold Cross Validation; Model evaluation metrics (Confusion Matrix, Precision, Recall, F_1-score, ROC-AUC) using `sklearn.metrics`.
UNIT V	**Unsupervised Machine Learning, Dimensionality & Deployment** **M**athematical concepts of Clustering (K-Means objective function, Distance metrics); Dimensionality Reduction principles (Principal Component Analysis variance maximization); Ethics, Bias, and Explainable AI (XAI) concepts. **I**mplementing K-Means and Hierarchical Clustering in Scikit-Learn; PCA implementation for feature compression; Pipeline construction (`sklearn.pipeline`); Model serialization (`pickle`, `joblib`); API wrapping preview using Flask/FastAPI.

Text Books

Text Books
1. Joel Grus, *Data Science from Scratch: First Principles with Python*, 2nd Edition, O'Reilly Media, 2019.
2. Jake VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*, 2nd Edition, O'Reilly Media, 2022.
3. Wes McKinney, *Python for Data Analysis*, 3rd Edition, O'Reilly Media, 2022.

Reference Books

Reference Books
1. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An Introduction to Statistical Learning with Applications in Python*, Springer, 2023.
2. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd Edition, O'Reilly Media, 2022.
3. Avrim Blum, John Hopcroft, and Ravindran Kannan, *Foundations of Data Science*, Cambridge University Press, 2020.

CSE403 – Database Management Systems

Course Information	
Course Code	CSE403
Course Name	Database Management Systems
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Basic Professional Course

Course Objectives

1. To understand the importance of a database system and how it is different from file management system.
2. To understand the overall architecture of database management system.
3. To study the physical and logical database designs, data independence, database modeling, relational, hierarchical, and network models.
4. To understand and use data manipulation language to query, update, and manage a database.
5. To develop an understanding of essential DBMS concepts such as: database security, integrity, transaction management and concurrency control, query optimization, Files and indexing technique.
6. To understand the some advanced topics of Database systems like distributed database, and intelligent database, Client/Server (Database Server), etc.
7. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Contents

Unit	Contents
Unit I	**Database System Architecture** Brief concept of Database system and how it is differed from File management system, Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). **Data Models** Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.
Unit II	**Relational Query Languages** Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS – MYSQL, ORACLE, DB2, SQL server. **Relational Database Design** Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.

Unit	Contents
Unit III	**Query Processing and Optimization** Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms. **Transaction Management** ACID Properties, Transaction and Schedules, Concurrent execution of schedules, Lock-based protocols, Cash recovery. **Concurrency Control** 2PL, Serializability and recoverability, Lock management, Deadlocks and deadlock prevention.
Unit IV	**Storage and Indexing** File organization and Indexing, Index data structure, different types of indexing techniques like, tree structured indexing, hash-based indexing. **Storage Data** Disk and files, Disk space management, Files of records, Page formats and record formats.
Unit V	**Advanced Topics** Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

Text Books

Text Books
1. *Database System Concepts*, 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

Reference Books

Reference Books
1. *Principles of Database and Knowledge – Base Systems*, Vol. 1 by J. D. Ullman, Computer Science Press.
2. *Fundamentals of Database Systems*, 5th Edition by R. Elmasri and S. Navathe, Pearson Education.
3. *Foundations of Databases*, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley.

Course Outcomes

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

1. For a given query write relational algebra expressions for that query and optimize the developed expressions.
2. For a given specification of the requirement design the databases using ER method and normalization.
3. For a given specification construct the SQL queries for Open source and Commercial DBMS MYSQL, ORACLE, and DB2.
4. For a given query optimize its execution using Query optimization algorithms.
5. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.
6. Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

CSE404 – Computer Organization & Architecture

Course Information	
Course Code	CSE404
Course Name	Computer Organization & Architecture
Category	Professional Core Course (PC)
Credits	L:3 T:1 P:0 C:4
Pre-Requisites	Digital Logic Design

Course Objectives

The student should be able to:

1. Understand the organization of a computer system from the user's perspective.
2. Understand various instruction formats and their execution in a computer system.

Course Outcomes

At the end of the course, the student should be able to:

1. Evaluate the performance of different computer architectures with respect to various parameters.
2. Study different pipeline hazards and their resolution techniques.
3. Analyze the performance of different Instruction Level Parallelism (ILP) techniques.
4. Identify cache and memory-related issues in multiprocessor systems.

Detailed Syllabus

Unit	Course Contents
UNIT I	**Introduction to Computer System** Representation of basic information, Computer types, Different functional units of a computer, Operational concept. Computer Organization and Computer Architecture and their differences. Performance of a Computer. Memory locations and addressing, Byte addressability, Big endian and Little endian assignment, Word alignment. Addressing modes and MIPS addressing. MIPS registers and instruction types. Operations of Computer Hardware, Operands of the Computer Hardware, Representing Instructions in the Computer, Logical Operations, Instructions for Making Decisions, Supporting Procedures in Computer Hardware.

Unit	Course Contents
UNIT II	**ALU Design** **High Speed Adder and Subtractor Design:** Ripple Carry Adder/Subtractor, Carry Look-Ahead Adder/Subtractor design, Multiplexer design, AND, OR, SLT and Overflow design. Design of an 8-bit ALU for ADD, SUB, AND, OR, RLL and RLR operations. **Multiplier Design:** Multiplication of positive numbers, Signed operand multiplication using Booth's Algorithm, Fast multiplier design, Carry Save Addition of Summands, Integer Division, Floating Point Number Representation and Floating Point Operations.
UNIT III	**Memory System** Basic concepts of memory, Semiconductor RAM memories, Read Only Memories (ROM), Speed–Size–Cost trade-off, Cache memories, Performance considerations, Virtual memory, Memory management requirements and Secondary storage.
UNIT IV	**Data Path Design and Control Design** Hardwired Control Unit and Microprogrammed Control Unit. MIPS datapath design for R-Type, I-Type and J-Type instructions and their corresponding hardwired control design.
UNIT V	**Pipeline** Overview of Pipelining, Pipelined Datapath and Control, Data Hazards: Forwarding versus Stalling, Control Hazards.

Text Books

Text Books
1. John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach*, 5th Edition, Morgan Kaufmann / Elsevier, 2012.

Reference Books

Reference Books
1. Kai Hwang and Faye A. Briggs, *Computer Architecture and Parallel Processing*, McGraw-Hill International Edition, 2000.
2. Dezső Sima, Terence Fountain and Peter Kacsuk, *Advanced Computer Architectures: A Design Space Approach*, Addison-Wesley, 2000.

CSE405 – Data Science Using Python Lab

Course Information	
Course Code	CSE405
Course Name	Data Science Using Python Lab
Category	Professional Core Course (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	CSE402 – Data Science Using Python

Course Objectives

The course aims to:

1. Provide hands-on experience in Python programming for data science applications.
2. Develop practical skills in data acquisition, cleaning, preprocessing and manipulation using Python.
3. Familiarize students with NumPy and Pandas for numerical computation and structured data analysis.
4. Enable students to create meaningful visualizations and perform exploratory data analysis using Python libraries.
5. Provide practical experience in developing and evaluating basic machine learning models using Scikit-learn.
6. Enable students to apply the complete data science workflow to real-world datasets through a mini-project.

Course Outcomes

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

1. Develop and execute Python programs for basic data science tasks.
2. Perform numerical computation and data manipulation using NumPy and Pandas.
3. Clean, preprocess and explore real-world datasets.
4. Create appropriate statistical and graphical visualizations using Matplotlib and Seaborn.
5. Implement basic regression, classification and clustering algorithms using Scikit-learn.
6. Develop a small data science application or project using a real-world dataset.

List of Laboratory Experiments

No.	Experiments
1	**Python Programming Environment and Fundamentals** Set up and configure a Python programming environment using Jupyter Notebook or Google Colab. Write Python programs demonstrating variables, data types, operators, conditional statements, loops, functions and basic exception handling.

No.	Experiments
2	**Python Data Structures and Functions** Implement programs using lists, tuples, sets and dictionaries. Develop user-defined functions and demonstrate modules, packages and basic file handling operations.
3	**NumPy Arrays and Numerical Computation** Create one-dimensional and multidimensional NumPy arrays. Perform indexing, slicing, reshaping, array operations, broadcasting, universal functions and vectorized computations.
4	**Data Manipulation using Pandas** Create and manipulate Pandas Series and DataFrames. Import datasets from CSV and Excel files and perform indexing, filtering, sorting, aggregation and basic statistical operations.
5	**Data Cleaning and Preprocessing** Identify and handle missing values, duplicate records and inconsistent data. Perform data type conversion, data transformation and basic data preprocessing using Pandas.
6	**Data Integration and Aggregation** Perform concatenation, merging, joining and grouping operations on multiple datasets. Use GroupBy, aggregation functions and pivot tables to summarize and analyze data.
7	**Exploratory Data Analysis** Perform exploratory data analysis on a real-world dataset using descriptive statistics, summary statistics, correlation analysis and outlier detection.
8	**Data Visualization using Matplotlib** Create line charts, bar charts, pie charts, histograms, scatter plots and box plots using Matplotlib. Customize labels, titles, legends and axes to communicate data effectively.
9	**Data Visualization using Seaborn** Create statistical visualizations using Seaborn, including heatmaps, pair plots, distribution plots, violin plots and count plots. Analyze relationships and patterns in datasets through visualization.
10	**Data Preprocessing and Feature Engineering** Prepare a dataset for machine learning by performing feature selection, feature transformation, encoding of categorical variables, normalization or standardization and train-test splitting.
11	**Regression using Scikit-learn** Implement Linear Regression and Polynomial Regression using Scikit-learn. Train the models on suitable datasets and evaluate their performance using appropriate regression metrics.
12	**Classification using Scikit-learn** Implement classification algorithms such as Logistic Regression, Decision Tree, K-Nearest Neighbours and Naïve Bayes. Evaluate the models using accuracy, precision, recall, F1-score and confusion matrix.
13	**Clustering and Unsupervised Learning** Implement K-Means clustering using Scikit-learn. Determine suitable clusters, visualize the resulting groups and analyze the clustering results.
14	**Integrated Data Science Workflow** Perform an end-to-end data science workflow involving data acquisition, data cleaning, exploratory data analysis, visualization, preprocessing, model development and evaluation using a suitable real-world dataset.

Text Books

Text Books
1. Wes McKinney, *Python for Data Analysis*, 3rd Edition, O'Reilly Media, 2022.
2. Jake VanderPlas, *Python Data Science Handbook*, O'Reilly Media, 2016.
3. Joel Grus, *Data Science from Scratch: First Principles with Python*, 2nd Edition, O'Reilly Media, 2019.
4. Andreas C. Müller and Sarah Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly Media, 2016.

Reference Books

Reference Books
1. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, 3rd Edition, O'Reilly Media, 2022.
2. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, *An Introduction to Statistical Learning with Applications in Python*, Springer, 2023.
3. Foster Provost and Tom Fawcett, *Data Science for Business*, O'Reilly Media, 2013.
4. Allen B. Downey, *Think Stats: Exploratory Data Analysis in Python*, 2nd Edition, O'Reilly Media, 2014.

CSE406 – Database Management Systems Lab

Course Information	
Course Code	CSE406
Course Name	Database Management Systems Lab
Category of Course	Professional Core Courses (PC)
Credits	L:0 T:0 P:4 C:2

Course Objectives

1. To provide a sound introduction to the creation of problem statements from real life situations.
2. To give a good formal foundation on the relational model of data and usage of Relational Algebra.
3. To introduce the concepts of basic SQL as a universal Database language.
4. To enhance knowledge to advanced SQL topics like embedded SQL, procedures connectivity through JDBC.
5. To enable the design of an efficient database using normalization concepts.
6. To enable students to be create indexes for databases for efficient retrieval.
7. To enable the student to experiment different transaction concept practically.
8. To provide a introduction to Use of host language interface with embedded SQL.

Experiments

No.	Experiments
1	Creating table, inserting data, updating table data, data record deletion, viewing data, modifying table structure, renaming and destroying table.
2	Arithmetic, logical operator, range searching, pattern matching, numeric function– scalar & group functions, string functions, Date function, table conversion functions.
3	Grouping data, join, sub-queries, union, intersection, minus clause, indexing, view, granting and revoking permissions.
4	Null value concept, primary key, and foreign key, unique, creating constraints, creating Indexes.
5	Introduction to PL/SQL – data type, branching, looping, simple problem solving using PL/SQL, Transaction concepts – commit, rollback, save point, introduction to cursor, parameterized cursor, locking.
6	Stored procedure and functions, package, trigger.
7	Use of host language interface with embedded SQL.
8	Use of user interfaces and report generation utilities typically available with RDBMS products.

Text Books

Text Books
1. *Database System Concepts*, 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2. *Fundamentals of Database Systems*, 5th Edition by R. Elmasri and S. Navathe, Pearson Education.

Reference Books

Reference Books
1. *Principles of Database and Knowledge – Base Systems*, Vol. 1 by J. D. Ullman, Computer Science Press.
2. *Foundations of Databases*, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley.

Course Outcomes

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

1. Construct problem definition statements for real life applications and implement a database for the same.
2. Design conceptual models of a database using ER modeling for real life applications and also construct queries in Relational Algebra.
3. Create and populate a RDBMS, using SQL.
4. Write queries in SQL to retrieve any type of information from a data base.
5. Analyze and apply concepts of normalization to design an optimal database.
6. Analyze and apply concepts of transactions.
7. Practically learn the concept of user interfaces and report generation utilities of RDBMS products.

CSE407 – IT Workshop I (Web Development)

Course Information	
Course Code	CSE407
Course Name	IT Workshop I (Web Development)
Category	Professional Core Course (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	Basic knowledge of computer fundamentals and programming

Course Objectives

The objectives of this laboratory course are:

1. To introduce students to the fundamental concepts and technologies of web development.
2. To develop practical skills in designing structured web pages using HTML5.
3. To familiarize students with CSS3 for styling, page layout, responsive design, and user-interface development.
4. To develop programming skills in JavaScript for creating interactive and dynamic web pages.
5. To introduce client-side form processing, validation, DOM manipulation, and browser-based web APIs.
6. To familiarize students with basic web-development tools, version control, and deployment of static websites.
7. To develop the ability to design and implement a complete responsive web application through a mini-project.

Course Outcomes

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

1. Develop structured and semantically meaningful web pages using HTML5.
2. Design responsive and user-friendly web interfaces using CSS3, Flexbox, Grid, and media queries.
3. Develop interactive client-side applications using JavaScript and DOM manipulation.
4. Implement client-side form validation and browser-based data storage.
5. Consume web services and process JSON data using JavaScript and the Fetch API.
6. Use basic version-control practices and deploy static web applications.
7. Design and implement a small web application integrating the major concepts covered in the laboratory.

List of Suggestive Laboratory Experiments

No.	Experiments
1	**Introduction to Web Development Environment** Set up a web development environment using a suitable code editor and web browser. Understand the structure of a web project, create an HTML document, and execute it using a browser. Explore browser Developer Tools.
2	**HTML5 Document Structure and Semantic Elements** Create web pages using HTML5 document structure and semantic elements such as `header`, `nav`, `section`, `article`, `aside`, and `footer`. Demonstrate headings, paragraphs, lists, hyperlinks, images, and multimedia elements.
3	**Tables and Forms using HTML5** Design web pages containing tables and forms. Use different HTML5 form controls, including text fields, password fields, radio buttons, checkboxes, select lists, text areas, date fields, email fields, and buttons.
4	**CSS Fundamentals and Web Page Styling** Apply CSS using inline, internal, and external stylesheets. Demonstrate selectors, colors, backgrounds, fonts, borders, margins, padding, dimensions, and the CSS box model.
5	**CSS Layout Techniques** Design web page layouts using CSS positioning, `display` properties, Flexbox, and Grid. Create structured layouts for headers, navigation bars, content sections, sidebars, and footers.
6	**Responsive Web Page Design** Develop responsive web pages using flexible layouts, relative units, media queries, responsive images, and mobile-first design principles. Test the design at different screen sizes using browser Developer Tools.
7	**JavaScript Fundamentals** Develop JavaScript programs demonstrating variables, constants, data types, operators, conditional statements, loops, functions, arrays, objects, and basic error handling.
8	**JavaScript Events and Interactive Web Pages** Develop interactive web pages using JavaScript events. Demonstrate events such as `click`, `change`, `mouseover`, `keydown`, and `submit`, and dynamically modify webpage content based on user actions.
9	**Document Object Model (DOM) Manipulation** Use JavaScript to access and manipulate HTML elements through the Document Object Model (DOM). Demonstrate element selection, modification of content and attributes, dynamic styling, creation and removal of elements, and event listeners.
10	**Client-Side Form Validation** Develop JavaScript-based validation for web forms. Validate user input such as name, email, phone number, password, age, and other appropriate fields. Display meaningful validation messages to users.
11	**JavaScript Objects, JSON and Local Storage** Create and manipulate JavaScript objects and JSON data. Demonstrate serialization and deserialization and use browser Local Storage to save and retrieve application data.

No.	Experiments
12	**Fetch API and Asynchronous JavaScript** Use the Fetch API to retrieve data from a web service. Demonstrate asynchronous programming using Promises and `async/await`, process JSON responses, and dynamically display retrieved information on a webpage.
13	**Introduction to Git and Web Project Management** Create and manage a Git repository for a web project. Perform basic operations such as repository initialization, staging, committing, branching, merging, and maintaining project versions.
14	**Static Website Development and Deployment** Design a responsive multi-page website using HTML5, CSS3, and JavaScript. Organize the project using appropriate folders and files and deploy the website using a suitable static hosting platform.
15	**Mini Project** Design and implement a small responsive web application integrating HTML5, CSS3, JavaScript, DOM manipulation, form validation, and appropriate browser APIs. Suggested projects include: • Student Information Portal • Department Website • Personal Portfolio Website • Online Quiz Application • Event Registration System • To-Do List Application • Student Attendance Interface • Online Book Catalogue • College Club/Society Website • Simple Expense Tracker The mini-project should include appropriate documentation, source code, testing, and demonstration.

Text Books

Text Books
1. Jon Duckett, *HTML and CSS: Design and Build Websites*, Wiley.
2. Jon Duckett, *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley.
3. Jennifer Niederst Robbins, *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, O'Reilly Media.
4. Marijn Haverbeke, *Eloquent JavaScript*, No Starch Press.

Reference Books

Reference Books
1. David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly Media.

2. Eric A. Meyer and Estelle Weyl, *CSS: The Definitive Guide*, O'Reilly Media.
3. Ethan Marcotte, *Responsive Web Design*, A Book Apart.
4. Steve Krug, *Don't Make Me Think, Revisited*, New Riders.

ASH401 – Management-I(Organisational Behaviour)

Course Information	
Course Code	ASH 401
Course Name	Management-I(Organisational Behaviour)
Category of Course	Humanities & Social Science including Management Courses
Credits	L:3 T:0 P:0 C:3

Objectives
The objective of the course is to orient the engineering students with the concepts and practical implications of Behavior, personality and attitude of individuals and groups in organization.

Course Outcomes
• The students will acquire the skills of understanding individual and group behavior, culture, attitude and personality. • The students will gain the knowledge of organizational behavior.

Course Contents

Unit	Contents
Unit-I	**Organisational Behaviour:** Concept and Emergence of OB Concept; Historical Background – Hawthorne Studies, Psychological foundations; Models of OrganisationalBehaviour, Challenges and Opportunities for Organisational Behavior; Ethics and Organisational Behaviour.
Unit–II	**Individual Behaviour:** Personality, Learning, Values and Attitudes, Perception, Learning Behaviourist, cognitive and social learning; Stress at work. Management's assumptions about people – McGregor's Theory X and Theory Y.
Unit–III	**Motivation:** Maslow's Need Hierarchy, Herzberg's Two Factors Theory, Vroom's Expectancy Theory; Theory of Intrinsic Motivation by Ken Thomas; Work – Designing for creating motivating Jobs.
Unit–IV	**Inter-personal Behaviour:** Interpersonal communication and Feedback, Feedback utilisation; Transactional Analysis (TA); Johari Window. Group Behaviour: Group Dynamics, Cohesiveness and Productivity; Group Decision Making; Organisational Politics.

Unit	Contents
Unit–V	**Leadership:** Concept and Styles; Fielder's Contingency Model; Leadership Effectiveness; Sources, patterns, levels, and types of conflict; Traditional and modern approaches to conflict; Functional and dysfunctional conflicts; Resolution of conflict. Organisational change- resistance and management.

Text Books

Text Books
1. Robbins, Stephen P. and Timothy A. Judge: *Organisational Behaviour*. Prentice-Hall, New Delhi.
2. Aswathappa, K: *OrganisationBehaviour*. Himalaya Publishing House, New Delhi.

Reference Books

Reference Books
1. Singh, K: *Organizational Behaviour: Text and Cases*. Pearson.
2. Pareek, U. and Khanna, S: *Understanding Organizational Behaviour*. Oxford University Press.
3. Sharma, R. A: *Organisational Theory and Behaviour*. Tata McGraw-Hill Publishing Co. Ltd.
4. Sekaran, Uma: *OrganisationalBehaviour: Text and Cases*. Tata McGraw-Hill Publishing Co. Ltd.
5. Singh, B. P. and T. N. Chhabra: *Organisation Theory and Behaviour*. DhanpatRai and Co. P. Ltd., New Delhi; 2000.

ASH402 – Environmental Science

Course Information	
Course Code	ASH402
Course Name	Environmental Science
Category of Course	Mandatory Non-Credit Courses
Credits	L:2 T:0 P:0 C:0 *(Non-Credit)*

Course Description

We as human being are not an entity separate from the environment around us rather we are a constituent seamlessly integrated and co-exist with the environment around us. We are not an entity so separate from the environment that we can think of mastering and controlling it rather we must understand that each and every action of ours reflects on the environment and vice versa. Ancient wisdom drawn from Vedas about environment and its sustenance reflects these ethos. There is a direct application of this wisdom even in modern times. Idea of an activity based course on environment protection is to sensitize the students on the above issues through following two type of activities.

(a) Awareness Activities:

i) Small group meetings about water management, promotion of recycle use, generation of less waste, avoiding electricity waste
ii) Slogan making event
iii) Poster making event
iv) Cycle rally
v) Lectures from experts

(b) Actual Activities:

i) Plantation
ii) Gifting a tree to see its full growth
iii) Cleanliness drive
iv) Drive for segregation of waste
v) To live some big environmentalist for a week or so to understand his work
vi) To work in kitchen garden for mess
vii) To know about the different varieties of plants
viii) Shutting down the fans and ACs of the campus for an hour or so

Course Contents

Unit	Contents
Unit I	**Multidisciplinary Nature of Environmental Studies** Definition, scope and importance; need for public awareness.

Unit	Contents
Unit II	**Natural Resources** Renewable and non-renewable resources: Natural resources and associated problems. **a) Forest resources:** Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forest and tribal people. **b) Water resources:** Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems. **c) Mineral resources:** Use and exploitation, environmental effects of extracting and using mineral resources, case studies. **d) Food resources:** World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies. **e) Energy resources:** Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources. Case studies. **f) Land resources:** Land as a resource, land degradation, man induced landslides, soil erosion and desertification. • Role of an individual in conservation of natural resources. • Equitable use of resources for sustainable lifestyles.
Unit III	**Ecosystems** • Concept of an ecosystem. • Structure and function of an ecosystem. • Producers, consumers and decomposers. • Energy flow in the ecosystem. • Ecological succession. • Food chains, food webs and ecological pyramids. • Introduction, types, characteristic features, structure and function of the following ecosystem: a. Forest ecosystem b. Grassland ecosystem c. Desert ecosystem d. Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

Unit	Contents
Unit IV	**Environmental Pollution** **Definition** Cause, effects and control measures of: a. Air pollution b. Water pollution c. Soil pollution d. Marine pollution e. Noise pollution f. Thermal pollution g. Nuclear hazards **Solid Waste Management:** Causes, effects and control measures of urban and industrial wastes. • Role of an individual in prevention of pollution. • Pollution case studies. • Disaster management: floods, earthquake, cyclone and landslides.
Unit V	**Social Issues and the Environment** • From Unsustainable to Sustainable development • Urban problems related to energy • Water conservation, rain water harvesting, watershed management • Resettlement and rehabilitation of people; its problems and concerns. Case Studies • Environmental ethics: Issues and possible solutions. • Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case Studies. • Wasteland reclamation. • Consumerism and waste products. • Environment Protection Act. • Air (Prevention and Control of Pollution) Act. • Water (Prevention and control of Pollution) Act • Wildlife Protection Act • Forest Conservation Act • Issues involved in enforcement of environmental legislation. • Public awareness.

Text/Reference Books

Text/Reference Books
1. *A Textbook of Environmental Studies*, D. K. Asthana, S Chand Publishing.
2. *Fundamental Concepts in Environmental Studies*, D. D. Mishra, S Chand Publishing.
3. *Environmental*, R. Rajagopalan, OUP India.
4. *Introduction to Environmental Engineering and Science*, Gilbert M. Masters and Wendell P. Ela, 3rd Edition, Pearson.

5. *Principles of Environmental Science: Inquiry & Applications - Inquiry and Applications*, Cunningham William, McGraw-Hill.

DETAILED SYLLABUS
Third Year – 5th Semester

Sl.	Type	Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE501	Software Engineering	3	1	0	4
2	PC	CSE502	Design and Analysis of Algorithms	3	0	0	3
3	PC	CSE503	Operating System	3	0	0	3
4	PC	CSE504	Computer Networks	3	0	0	3
5	PC	CSE505	Design and Analysis of Algorithms Lab	0	0	4	2
6	PC	CSE506	Operating System Lab	0	0	4	2
7	PC	CSE507	Computer Networks Lab	0	0	4	2
8	PR	CSE508	Summer Training (4 Weeks)	0	0	0	0
9	HC	ASH501	Management II (Project Management / Operations Research and Industrial Management)	3	0	0	3
10	MC	ASH502	Essence of Indian Knowledge Tradition/Constitution of India	2	0	0	0
						Total Credits	**22**

CSE501 – Software Engineering

Course Information	
Course Code	CSE501
Course Name	Software Engineering
Category of Course	Professional Core Course (PC)
Credits	L:3 T:1 P:0 C:4
Pre-Requisites	NIL

Course Objectives
1. To discuss the process for developing large software. 2. To analyse and model a particular system. 3. To develop alternative solutions for the system. 4. To implement, test and validate a systems design.

Course Contents

Unit	Contents
Unit I	**Overview of System Analysis & Design** Overview of System Analysis & Design, Business System Concept, System Development Life Cycle, Waterfall Model, Spiral Model, Feasibility Analysis, Technical Feasibility, Cost-Benefit Analysis, COCOMO model, Function Point Analysis (FPA).
Unit II	**System Requirement Specification and System Analysis** System Requirement Specification, System analysis – DFD, Data Dictionary, ER diagram, Process Organization & Interactions. **System Design:** Problem Partitioning, Top-Down & Bottom-Up design; Decision tree, decision table and structured English; Functional vs. Object-Oriented approach.
Unit III	**Coding & Documentation** Structured Programming, OO Programming, Information Hiding, Reuse, System Documentation.
Unit IV	**Testing** Levels of Testing, White & Black box testing, Integration Testing, structural testing, Test case Specification, Reliability Assessment, Validation & Verification Metrics, Monitoring & Control.
Unit V	**Software Project Management** Project Scheduling, Staffing, Software Configuration Management, Quality Assurance, Project Monitoring. **CASE TOOLS:** Concepts, use and application. Software reliability and quality management.

Text Books

Text Books
1. Rajib Mall, *Fundamentals of Software Engineering*, 3rd ed., PHI.

Reference Books

Reference Books
1. R. G. Pressman, *Software Engineering*, TMH.
2. Behforooz, *Software Engineering Fundamentals*, OUP.

Course Outcomes
Upon completion of this course, students will be able to: 1. Discuss the process for developing large software. 2. Analyse and model a particular system. 3. Develop alternative solutions for the system. 4. Implement, test and validate a systems design.

CSE502 – Design and Analysis of Algorithms

Course Information	
Course Code	CSE502
Course Name	Design and Analysis of Algorithms
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Basic knowledge of introductory courses on mathematics, Programming, and Data Structures.

Course Objectives

1. Introduce students to the basic concept of algorithms in computing, analyzing algorithms, and designing algorithms.
2. Students learn about the asymptotic notation of algorithms.
3. Students should be able to write correctness of proofs for algorithms. Also, able to analyze the efficiency of algorithms based on asymptotic complexity.
4. Students should be able to demonstrate different modeling of problem-solving like a graph, data structures, decomposing the problem.
5. Learn about the different techniques of algorithms like divide-and-conquer, greedy, dynamic programming.
6. Students will be able to synthesize efficient algorithms in a given engineering problem.

Course Contents

Unit	Contents
Unit I	**Introduction of Algorithms** Introduction, Motivation, the role of algorithms in computing. **Analyzing of Algorithms** Model of Computation like RAM, TM, etc., space and time complexity, asymptotic notation, functions, and running time are applied in well-known algorithms like heap sort, search algorithms, etc. **Designing Algorithms** Definition of recursion, use, and limitation, Examples of Towers Hanoi, Tail recursion, etc., an overview of designing techniques.

Unit	Contents
Unit II	**Divide and Conquer** Basic concept, element of dynamic programming, use, Examples – Quick sort, Merge sort, Binary search, the maximum-subarray problem, Strassen's algorithm for matrix multiplication, etc., Methods for solving recurrences. **Dynamic Programming** Basic concept, use, Examples – matrix-chain multiplication, All pair shortest paths, Single-source shortest path, Longest common subsequence Traveling Salesman problem etc. **Branch and Bound** Basic concept, Least cost search, use, Example – The 15-puzzle problem, 0/1 knapsack problem, Traveling salesman problem etc.
Unit III	**Backtracking Method** Basic concept, use, Examples – 8-Queens problem, Graph coloring problem, Hamiltonian, knapsack problem, etc. **Greedy Method** Basic concept, use, Examples – Knapsack problem, Job sequencing with deadlines, Huffman Coding, Matroids, task-scheduling problems, minimum spanning tree (Prim's and Kruskal's algorithms). **Lower Bound Theory** Comparison trees based on searching, sorting, and selection, Lower Bound techniques through reduction.
Unit IV	**Disjoint Set Manipulation** Set manipulation algorithm like UNION-FIND, union by rank, Path. **Graph Algorithms** Properties of graphs and graph traversal algorithms: BFS and DFS, Minimum Spanning Trees, Graph traversal Shortest Path problems, Maximum Flow problems etc.
Unit V	**NP-Completeness** Notion of NP-completeness: P class, NP-hard class, NP-complete class, Circuit Satisfiability problem, Clique Decision Problem, etc. **Approximation Algorithms** Necessity of approximation scheme, performance guarantee, Polynomial time approximation schemes: 0/1 knapsack problem, Traveling-salesman problem, Vertexcover Problem.

Text Books

Text Books
1. *Introduction to Algorithms*, 3rd Edition, T. H. Cormen, C. E. Leiserson, R. L. Rivest, and Clifford Stein, published by PHI Learning Private Limited (Original edition published by the MIT Press, Cambridge, MA, USA), Copyright © 2011.
2. *Fundamental of Computer Algorithms*, 2nd Edition, E. Horowitz, S. Sahni, and S. Rajasekaran, published by Universities Press (India) Private Limited, Copyright © 2008, 2010.

Reference Books

Reference Books
1. *Algorithm Design*, 1st Edition, Jon Kleinberg, and Eva Tardos, published by Pearson Education Limited, Copyright © 2014.
2. *Algorithms*, 1st Edition, S. Dasgupta, C. Papadimitriou, and U. Vazirani, published by McGraw-Hill Education, Copyright © 2008.

Course Outcomes

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

1. Students will be able to apply the concept and design strategies to algorithm design.
2. Students will be able to analyze the efficiency of algorithms based on space and time complexity theory.
3. Students will be capable of understanding the different type algorithm design techniques, and also learned the concept of which design technique is more suited for finding the solution of a given problem.
4. Students will be able to synthesize the efficient algorithm in a given engineering problem.

CSE503 – Operating System

Course Information	
Course Code	CSE503
Course Name	Operating System
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	

Course Objectives

1. To learn the mechanisms of OS to handle processes and threads and their communication.
2. To learn the mechanisms involved in memory management in contemporary OS.
3. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.
4. To know the components and management aspects of concurrency management.
5. To learn to implement simple OS mechanisms.

Course Contents

Unit	Contents
Unit I	**Introduction** Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS – Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.
Unit II	**Processes** Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. **Thread:** Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads. **Process Scheduling** Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.
Unit III	**Inter-process Communication** Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer\Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc. **Deadlocks** Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, and Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

Unit	Contents
Unit IV	**Memory Management** Strategies –Contiguous and Non-Contiguous allocation – Paging Schemes, Paging Hardware, Segmentation - **Virtual memory** Management –Demand Paging, Demand Segmentation , Page Replacement Policies
Unit V	**I/O Hardware** I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms. **File Management** Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. **Disk Management** Disk structure, Disk scheduling – FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

Text Books

Text Books
1. *Operating System Concepts Essentials*, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. *Operating Systems: Internals and Design Principles*, 5th Edition, William Stallings, Prentice Hall of India.

Reference Books

Reference Books
1. Dhamdhere, D. M. (2017). Operating Systems: A Concept-based Approach (3rd ed.). McGraw Hill Education.
2. Andrew S. Tanenbaum, "Modern Operating Systems", 4/E, Pearson Publications, 2014

Course Outcomes
• The skill that a student will acquire. • The knowledge (Theoretical/applied/both) the student will gain.

CSE504 – Computer Networks

Course Information	
Course Code	CSE504
Course Name	Computer Networks
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	

Course Objectives

- To develop an understanding of modern network architectures from a design and performance perspective.
- To introduce the students to the major concepts involved in Wide-Area Networks (WANs), Local Area Networks (LANs) and Wireless LANs (WLANs).
- To provide an opportunity to do network programming.
- To provide a WLAN measurement ideas.

Course Contents

Unit	Contents
Unit I	**Data Communication and LAN** Data communication Components: Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, TCP/IP protocol suite, Transmission Media, Network performance metrics. **LAN:** Wired LAN, Ethernet Standards, Wireless LANs, Virtual LAN, Bluetooth. **Techniques for Bandwidth Utilization:** Multiplexing – Frequency division, Time division and Wave division, Concepts on spread spectrum.
Unit II	**Data Link Layer and Medium Access Sub Layer** Error Detection and Error Correction – Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols – Stop and Wait, Go back–N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols – Pure ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA, Spanning tree Protocol, VLAN Tagging.
Unit III	**Network Layer** Switching, Logical addressing – IPV4, IPV6; Subnetting, Supernetting, CIDR, ARP, DHCP, ICMP, NAT, Delivery, Forwarding and Unicast Routing protocols (RIP, OSPF), BGP.

Unit	Contents
Unit IV	**Transport Layer** Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Socket, SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm. QUIC.
Unit V	**Application Layer and Modern Networks** Domain Name Space (DNS), DDNS, SMTP, WWW, HTTPS, SNMP, SSH, TLS/SSL, Introduction to Software Defined Network (SDN), Bluetooth, Introduction to Mobile and Wireless Computing, Emerging networking trends.

Text Books

Text Books
1. *Computer Networks*, 6th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. *Data and Computer Communication*, 11th Edition, William Stallings, Pearson Prentice Hall India.
3. *Computer Networking: A Top-Down Approach*, 9th Edition, James F. Kurose & Keith W. Ross.

Reference Books

Reference Books
1. *Data Communication and Networking*, 6th Edition, Behrouz A. Forouzan, McGraw-Hill.
2. *Internetworking with TCP/IP*, Volume 1, 6th Edition, Douglas Comer, Prentice Hall of India.
3. *TCP/IP Illustrated*, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

Course Outcomes
Upon completion of this course, students will be able to: 1. Explain the functions of the different layer of the OSI Protocol. 2. Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block. 3. For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) design it based on the market available component. 4. For a given problem related TCP/IP protocol develop the network programming. 5. Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

CSE505 – Design and Analysis of Algorithms Lab

Course Information	
Course Code	CSE505
Course Name	Design and Analysis of Algorithms Lab
Category of Course	Professional Core Course (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	CSE502 – Design and Analysis of Algorithms

Course Objectives

1. To develop practical skills in implementing fundamental algorithms and analyzing their computational complexity.
2. To experimentally compare the performance of algorithms with their theoretical asymptotic complexity.
3. To implement and apply major algorithm design paradigms including divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound.
4. To implement fundamental graph algorithms and supporting data structures such as disjoint-set structures.
5. To develop practical understanding of computationally difficult problems, approximation algorithms, and algorithmic trade-offs.
6. To enable students to select and implement appropriate algorithmic techniques for solving computational and engineering problems.

List of Experiments

No.	Experiments
1	**Empirical Analysis of Algorithms** Implement representative iterative and recursive algorithms and experimentally determine their execution time for different input sizes. Analyze and compare the experimental results with theoretical complexity using O-notation, Ω-notation, and Θ-notation. Plot execution time against input size and discuss best-case, average-case, and worst-case behavior.
2	**Searching and Sorting Algorithms** Implement and compare Linear Search, Binary Search, Insertion Sort, Selection Sort, and Heap Sort. Analyze their time and space complexities and experimentally compare their performance for different input sizes.
3	**Divide-and-Conquer Algorithms** Implement and analyze Merge Sort, Quick Sort, Binary Search, and the Maximum Subarray Problem. Compare the empirical performance of the algorithms and relate the results to their recurrence relations and asymptotic complexities.

No.	Experiments
4	**Strassen's Matrix Multiplication** Implement Strassen's matrix multiplication algorithm and compare its execution time with conventional matrix multiplication for different matrix sizes.
5	**Greedy Algorithms for Optimization Problems** Implement the following greedy algorithms: Fractional Knapsack, Job Sequencing with Deadlines, and Activity Selection / Task Scheduling. Analyze the greedy-choice property and compare the solutions obtained for appropriate problem instances.
6	**Minimum Spanning Tree Algorithms** Implement and compare Prim's Algorithm and Kruskal's Algorithm for finding the Minimum Spanning Tree of a weighted undirected graph. Use suitable graph representations and experimentally compare their performance for sparse and dense graphs.
7	**Shortest Path Algorithms** Implement Dijkstra's Algorithm for finding single-source shortest paths in graphs with non-negative edge weights. Analyze the algorithm using different graph representations and input sizes.
8	**Dynamic Programming for Optimization Problems** Implement the following dynamic programming algorithms: 0/1 Knapsack, Matrix Chain Multiplication, and Longest Common Subsequence (LCS). For each problem, demonstrate the construction of the dynamic programming table and determine the optimal solution.
9	**Dynamic Programming for Shortest Paths** Implement the Floyd–Warshall Algorithm for the All-Pairs Shortest Path Problem. Analyze its time and space complexity and compare its applicability with single-source shortest-path algorithms.
10	**Backtracking Algorithms** Implement and analyze the N-Queens Problem, Sum of Subsets Problem, and Graph Coloring Problem using the backtracking technique. Study the state-space tree and analyze the effect of increasing the problem size.
11	**Branch-and-Bound Algorithms** Implement a branch-and-bound solution for the Travelling Salesperson Problem (TSP). Demonstrate state-space representation, branching, bounding function, selection of the next promising node, and pruning of non-promising nodes.
12	**Disjoint Set Operations and Kruskal's Algorithm** Implement the Union-Find / Disjoint Set data structure with Make-Set, Find, Union, Union by Rank, and Path Compression. Apply the structure to Kruskal's Minimum Spanning Tree algorithm and analyze the performance improvement obtained through union by rank and path compression.
13	**Graph Traversal and Network Algorithms** Implement and analyze Breadth-First Search (BFS) and Depth-First Search (DFS). Use the algorithms for connectivity analysis and cycle detection. Implement the algorithms using both adjacency-matrix and adjacency-list representations and compare their computational requirements.
14	**Maximum Flow and Computationally Difficult Problems** Implement the Ford–Fulkerson Algorithm for the Maximum Flow problem using augmenting paths. Also demonstrate a small-scale implementation of an NP-complete decision problem such as the Clique Decision Problem or Circuit Satisfiability Problem. Discuss the distinction between polynomial-time and computationally difficult problems.

No.	Experiments
15	**Approximation Algorithms and Algorithmic Trade-offs** Implement an approximation algorithm for one or more computationally difficult problems, such as Vertex Cover, Travelling Salesperson Problem, or 0/1 Knapsack. Evaluate the approximate solution against an optimal or known solution for small instances and determine the approximation/performance guarantee where applicable.

Course Outcomes

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

1. Implement fundamental algorithms and experimentally analyze their time and space complexities.
2. Apply divide-and-conquer and greedy techniques to solve searching, sorting, optimization, and graph problems.
3. Design dynamic programming solutions for combinatorial optimization and shortest-path problems.
4. Implement backtracking and branch-and-bound algorithms for state-space search and optimization problems.
5. Implement graph algorithms and disjoint-set operations for solving network and graph-based problems.
6. Develop and evaluate approximation solutions for computationally difficult problems.
7. Select, implement, and evaluate appropriate algorithmic techniques for solving real-world and engineering problems.

CSE506 – Operating System Lab

Course Information	
Course Code	CSE506
Course Name	Operating System Lab
Category of Course	Professional Core Courses (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	NIL

Course Objectives
This lab complements the operating systems course. Students will gain practical experience with designing and implementing concepts of operating systems such as system calls, CPU scheduling, process management, memory management, file systems and deadlock handling using C language in Linux environment.

List of Experiments

No.	Experiments
1	**Shell Scripting and Shell Programming** Overview of Shell scripting and shell programming.
2	Write a C program to simulate the following CPU Scheduling algorithms to find turnaround time and waiting time: a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority Scheduling
3	Write a C program to simulate Banker's algorithm for the purpose of deadlock avoidance.
4	Write a C program to simulate Peterson's software solution for Race condition.
5	Write a C program to simulate the following contiguous memory allocation techniques: a) Worst-fit b) Best-fit c) First-fit
6	Write a C program to simulate the following page replacement algorithms: a) FIFO b) LRU c) LFU
7	Write a C program to simulate the following file organization techniques: a) Single level directory b) Two level directory c) Hierarchical

Course Outcomes

Upon the completion of Operating Systems practical course, the student will be able to:

1. Understand and implement basic services and functionalities of the operating system using system calls.
2. Use modern operating system calls and synchronization libraries in software/hardware interfaces.
3. Understand the benefits of thread over process and implement synchronized programs using Multithreading concepts.
4. Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
5. Implement memory management schemes and page replacement schemes.
6. Simulate file allocation and organization techniques.
7. Understand the concepts of deadlock in operating systems and implement them in multi-programming System.

CSE507 – Computer Networks Lab

Course Information	
Course Code	CSE507
Course Name	Computer Networks Lab
Category of Course	Professional Core Courses (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	C/C++ Programming

Course Objectives

- To introduce Network related commands and configuration files in Linux Operating System.
- To introduce tools for Network Simulation.
- To introduce Socket programming for client server application.

List of Experiments

No.	Experiments
1	**Implementation of Error Detection / Error Correction Techniques** Implement error detection and error correction techniques used in computer networks.
2	**Implementation of Data Link Layer Flow Control Techniques** Implement data link layer flow control techniques.
3	**Study of Socket Programming and Client–Server Model** Study socket programming and the client–server model.
4	**Socket Programming for Network Commands** Write a socket program for Echo/Ping/Talk commands.
5	**Simulation of Routing Protocols** Simulate different routing protocols like RIP, OSPF, and EIGRP.
6	**Simulation of Network Protocols** Simulate other protocols like NAT, VLAN, and ACL etc.
7	**Encryption and Decryption** Implement encryption and decryption.

Text Books / Reference Books

Text Books / Reference Books

1. *TCP/IP Illustrated*, Volume 1, W. Richard Stevens, and Addison-Wesley, United States of America.

Course Outcomes
Upon completion of this laboratory course, students will be able to: 1. Use network related commands and configuration files in Linux Operating System. 2. For a given problem related TCP/IP protocol develop the network programming. 3. Configure different protocols using open source available software and tools. 4. Analyze network traffic using network monitoring tools.

ASH501 – Management II (Project Management / Operations Research and Industrial Management)

ASH501 – Operations Research and Industrial Management

Course Information	
Course Code	ASH501
Course Name	Operations Research and Industrial Management
Category of Course	Humanities & Social Science including Management Courses
Credits	L:3 T:0 P:0 C:3

Course Contents

Unit	Contents
Unit I	System concepts, system approach, Linear programming problems, Mathematical formulation, Graphical solution, Simplex method; Degeneracy and Duality in linear programming.
Unit II	Transportation problems, Assignment problems, Decision analysis.
Unit III	Waiting line problems, Project Management by PERT/CPM; Inventory control.
Unit IV	Mathematical models of physical systems. Modeling of systems and Computer Simulation.
Unit V	Advanced Computer Programming Techniques: Integer Programming, Dynamic Programming.

Text Books

Text Books
1. L. C. Jhamb, *Industrial Management*, Vol. 1, EPH.
2. S. N. Chary, *Productions and Operations Management*, TMH.
3. Hamdy A. Taha, *Operations Research*, Pearson, Global Edition.

References

References
1. Phillip Kotler, *Marketing Management*, Prentice Hall/Pearson Education.
2. Joseph Monks, *Productions and Operations Management*, TMH.

ASH501 – Project Management

Course Information	
Course Code	ASH501
Course Name	Project Management
Category of Course	Humanities and Social Sciences including Management Courses
Credits	L:3 T:0 P:0 C:3

Course Objectives

The objectives of this course are to:

1. Make the learners understand the concepts of Project Management for planning to execution of projects.
2. Make them understand the feasibility analysis in Project Management and network analysis tools for cost and time estimation.
3. Enable them to comprehend the fundamentals of Contract Administration, Costing and Budgeting.
4. Make them capable to analyze, apply and appreciate contemporary project management tools and methodologies in Indian context.

Course Contents

Unit	Contents
Unit I	**Introduction to Project Management** Characteristics of projects, Definition and objectives of Project Management, Stages of Project Management, Project Planning Process, Establishing Project organization.
Unit II	**Work Definition and Project Planning** Defining work content, Time Estimation Method, Project Cost Estimation and budgeting, Project Risk Management, Project scheduling and Planning Tools: Work Breakdown Structure, LRC, Gantt charts, CPM/PERT Networks.
Unit III	**Project Planning and Resource Management** Developing Project Plan (Baseline), Project cash flow analysis, Project scheduling with resource constraints: Resource Levelling and Resource Allocation. **Time Cost Trade-off:** Crashing Heuristic.
Unit IV	**Project Implementation and Control** Project Implementation: Project Monitoring and Control with PERT/Cost, Computers applications in Project Management, Contract Management, Project Procurement Management.
Unit V	**Post-Project Analysis** Post-Project Analysis.

Course Outcomes
On completion of this course, the learners will be able to: 1. Understand project characteristics and various stages of a project. 2. Understand the conceptual clarity about project organization and feasibility analyses – Market, Technical, Financial and Economic. Execution Control. 3. Analyze the learning and understand techniques for Project planning, scheduling and Execution Control. 4. Apply the risk management plan and analyse the role of stakeholders. 5. Understand the contract management, Project Procurement, Service level Agreements and productivity. 6. Understand the How Subcontract Administration and Control are practiced in the Industry.

Text Books

Text Books
1. Shtub, Bard and Globerson, *Project Management: Engineering, Technology, and Implementation*, Prentice Hall, India.
2. Horald Kerzner, *Project Management: A Systemic Approach to Planning, Scheduling and Controlling*, CBS Publishers, 2002.
3. John M. Nicholas, *Project Management for Business and Technology: Principles and Practice*, Prentice Hall, India, 2002.

References

References
1. Wiest and Levy, *Management Guide to PERT/CPM*, Prentice Hall, India.
2. S. Choudhury, *Project Scheduling and Monitoring in Practice.*
3. P. K. Joy, *Total Project Management: The Indian Context*, Macmillan India Ltd.
4. N. J. Smith (Ed.), *Project Management*, Blackwell Publishing, 2002.
5. Robert K. Wysocki, Robert Back Jr. and David B. Crane, *Effective Project Management*, John Wiley, 2002.
6. Jack R. Meredith and Samuel J. Mantel, *Project Management: A Managerial Approach*, John Wiley, 2000.

ASH502 – Essence of Indian Knowledge Tradition / Constitution of India

ASH502 – Constitution of India

Course Information	
Course Code	ASH502
Course Name	Constitution of India
Category of Course	Mandatory Non-Credit Courses
Credits	L:2 T:0 P:0 C:0

Course Description

The Constitution of India is the supreme law of India. Parliament of India cannot make any law which violates the Fundamental Rights enumerated under Part III of the Constitution. The Parliament of India has been empowered to amend the Constitution under Article 368, however, it cannot use this power to change the "basic structure" of the Constitution, which has been ruled and explained by the Supreme Court of India in its historical judgments.

The Constitution of India reflects the idea of "Constitutionalism" – a modern and progressive concept historically developed by the thinkers of "liberalism" – an ideology which has been recognized as one of the most popular political ideology and result of historical struggles against arbitrary use of sovereign power by state.

The historic revolutions in France, England, America and particularly European Renaissance and Reformation movement have resulted into progressive legal reforms in the form of "constitutionalism" in many countries. The Constitution of India was made by borrowing models and principles from many countries including United Kingdom and America.

The Constitution of India is not only a legal document but it also reflects social, political and economic perspectives of the Indian Society. It reflects India's legacy of "diversity". It has been said that Indian constitution reflects ideals of its freedom movement, however, few critics have argued that it does not truly incorporate our own ancient legal heritage and cultural values.

No law can be "static" and therefore the Constitution of India has also been amended more than one hundred times. These amendments reflect political, social and economic developments since the year 1950.

The Indian judiciary and particularly the Supreme Court of India has played an historic role as the guardian of people. It has been protecting not only basic ideals of the Constitution but also strengthened the same through progressive interpretations of the text of the Constitution. The judicial activism of the Supreme Court of India and its historic contributions has been recognized throughout the world and it gradually made it "as one of the strongest court in the world".

Course Contents

No.	Course Content
1	Meaning of the constitution law and constitutionalism.
2	Historical perspective of the Constitution of India.
3	Salient features and characteristics of the Constitution of India.
4	Scheme of the fundamental rights.
5	The scheme of the Fundamental Duties and its legal status.
6	The Directive Principles of State Policy – Its importance and implementation.
7	Federal structure and distribution of legislative and financial powers between the Union and the States.
8	Parliamentary Form of Government in India – The constitution powers and status of the President of India.
9	Amendment of the Constitutional Powers and Procedure.
10	The historical perspectives of the constitutional amendments in India.
11	Emergency Provisions: National Emergency, President Rule, Financial Emergency.
12	Local Self Government – Constitutional Scheme in India.
13	Scheme of the Fundamental Right to Equality.
14	Scheme of the Fundamental Right to certain Freedom under Article 19.
15	Scope of the Right to Life and Personal Liberty under Article 21.

Suggested Books

Suggested Books
1. Madhav Khosla, *The Indian Constitution*, Oxford University Press, New Delhi, 2012.
2. Brij Kishore Sharma, *Introduction to the Indian Constitution*, PHI, New Delhi, latest edition.

ASH502 – Essence of Indian Knowledge Tradition

Course Information	
Course Code	ASH502
Course Name	Essence of Indian Knowledge Tradition
Category of Course	Audit Course
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	NIL

Course Objectives

The objectives of this course are to:

1. Impart basic principles of thought process, reasoning and inference to identify the roots and details of some of the contemporary issues faced by our nation and explore possible solutions to these challenges by drawing insights from our past.
2. Enable the students to understand the importance of their surroundings and encourage them to contribute towards sustainable development.
3. Sensitize students towards issues related to Indian culture, tradition and its composite character.
4. Make students aware of holistic lifestyles, Yogic science and wisdom capsules available in Sanskrit literature that are relevant to modern society amid rapid technological advancements and societal disruptions.
5. Acquaint students with the Indian Knowledge System, the Indian perspective of the modern scientific worldview, and the basic principles of Yoga and holistic healthcare systems.

Course Contents

Unit	Contents
Unit I	**Society, State and Polity in India** **State in Ancient India:** Evolutionary Theory, Force Theory, Mystical Theory and Contract Theory; stages of State Formation in Ancient India; Kingship; Council of Ministers; Administration; Political Ideals in Ancient India; conditions for the welfare of societies; and the Seven Limbs of the State. **Society in Ancient India:** Purusārtha; Varnāshrama System; Āshrama or the Stages of Life; Marriage; understanding gender as a social category; representation of women in historical traditions; challenges faced by women; four-class classification; and slavery.

Unit	Contents
Unit II	**Indian Literature, Culture, Tradition and Practices** Evolution of scripts and languages in India: Harappan Script and Brahmi Script; the Vedas; the Upanishads; the Ramayana and the Mahabharata; Puranas; Buddhist and Jain literature in Pali, Prakrit and Sanskrit; Kautilya's Arthashastra; famous Sanskrit authors; Telugu literature; Kannada literature; Malayalam literature; Sangam literature; Northern Indian languages and literature; Persian and Urdu literature; and Hindi literature.
Unit III	**Indian Religion, Philosophy and Practices** Pre-Vedic and Vedic Religion; Buddhism; Jainism; six systems of Indian Philosophy; Shankaracharya; various philosophical doctrines; other heterodox sects; Bhakti Movement; Sufi Movement; socio-religious reform movements of the 19th century; and modern religious practices.
Unit IV	**Science, Management and Indian Knowledge System** Astronomy in India; Chemistry in India; Mathematics in India; Physics in India; Agriculture in India; Medicine in India; Metallurgy in India; Geography; Biology; Harappan Technologies; Water Management in India; Textile Technology in India; Writing Technology in India; Pyrotechnics in India; Trade in Ancient India; and India's dominance up to pre-colonial times.
Unit V	**Cultural Heritage and Performing Arts** Indian Engineering and Architecture; Engineering and Architecture in Ancient India; Sculptures; Seals; Coins; Pottery; Puppetry; Dance; Music; Theatre and Drama; Painting; Martial Arts Traditions; Fairs and Festivals; current developments in Arts and Culture; India's Cultural Contribution to the World; and Indian Cinema.

Course Outcomes
On completion of this course, the learners will be able to: 1. Understand and explain the fundamental concepts of Indian traditional knowledge and its relevance to contemporary society. 2. Connect Indian knowledge traditions with modern scientific perspectives and contemporary challenges. 3. Develop an understanding of the social, political and cultural evolution of India through its historical traditions and knowledge systems. 4. Appreciate the contributions of Indian philosophy, religion, literature, science and technology to the development of human knowledge. 5. Recognize the relevance of Yoga, holistic lifestyles and traditional knowledge systems in promoting individual and societal well-being. 6. Appreciate India's cultural heritage and performing arts and evaluate India's contributions to the world.

Suggested Text & Reference Books

Suggested Text & Reference Books
1. V. Sivaramakrishna (Ed.), *Cultural Heritage of India – Course Material*, Bharatiya Vidya Bhavan, Mumbai, 5th Edition, 2014.

2. S. Baliyan, *Indian Art and Culture*, Oxford University Press, India.
3. Swami Jitatmanand, *Modern Physics and Vedanta*, Bharatiya Vidya Bhavan.
4. Romila Thapar, *Readings in Early Indian History*, Oxford University Press, India.
5. Fritjof Capra, *The Tao of Physics*.
6. Fritjof Capra, *The Web of Life*.
7. V. N. Jha (English Translation), *Tarkasangraha of Annam Bhatta*, International Chinmaya Foundation, Velliarnad, Ernakulam.
8. *Yoga Sutra of Patanjali*, Ramakrishna Mission, Kolkata.
9. G. N. Jha (English Translation), Ed. R. N. Jha, *Yoga-Darshanam with Vyasa Bhashya*, Vidyanidhi Prakasham, Delhi, 2016.
10. R. N. Jha, *Science of Consciousness, Psychotherapy and Yoga Practices*, Vidyanidhi Prakasham, Delhi, 2016.
11. P. R. Sharma (English Translation), *Shodashang Hridayam*.
12. Basham, A. L., *The Wonder That Was India*, 34th Impression, New Delhi, Rupa & Co.
13. Sharma, R. S., *Aspects of Political Ideas and Institutions in Ancient India*, Fourth Edition, Delhi, Motilal Banarsidass.

DETAILED SYLLABUS
Third Year – 6th Semester

Sl.	Type	Code	Course Title	Hours/Week			Credits
				L	T	P	
1	EC	CSE601	Research Methodology and Technical Writing	3	0	0	3
2	PC	CSE602	Compiler Design	3	0	0	3
3	PC	CSE603	Machine Learning	3	0	0	3
4	PE	CSE604(#)	Elective I	3	0	0	3
5	OE	CSE605(#)	Open Elective I	3	0	0	3
6	PC	CSE606	Compiler Design Lab	0	0	4	2
7	PC	CSE607	Machine Learning Lab	0	0	4	2
8	PC	CSE608	IT Workshop II (Mobile Application Development)	0	0	4	2
9	PR	CSE609	Minor Project	0	0	6	3
						Total Credits	**24**

Note: Students will have to take up **Mandatory Internship/Apprenticeship** of 8 weeks (80 hrs) at industry/organizations/research institutes of repute after the completion of 6th semester, that will be accredited in 7th semester.

CSE601 – Research Methodology and Technical Writing

Course Information	
Course Code	CSE601
Course Name	Research Methodology and Technical Writing
Category of Course	Engineering Core Courses (EC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	NIL

Course Objectives

1. To know the basics of how research problems are defined, research methods are adopted and/or developed, research is undertaken, and how research results are communicated to the peers.
2. To know the research methods specific to the field of computer science.

Syllabus

Unit	Contents	Hours
Unit I	**Underpinnings of Research** Introduction: Novelty, Originality, Organized Method of Investigation, and Communication of Problem, Data, Method, and Results for Peer Group Verification; Paradigm and paradigm shift. **Types of Research:** Theoretical, Empirical, Experimental, and Design and Characterization of New Materials, Components, Processes, and Systems. **Broad Objectives of Research:** Problem-oriented Defining Problems and Problem Issues, Analysing Data, Predicting, and Designing; Technique-oriented—Improving Performance and Expanding Scope. **Modes of Inquiry and Inquiring Systems:** Hypothetico-deductive and Empirical-inductive modes; Scientific Method; and Inquiring Systems of Locke, Kant, Leibnitz, Hegel, and Singer. Continuum of Connections among Facts, Data, Laws, Hypotheses, Theory, Models, and Experiments; Criteria of a Theory. Research Topic, Problem, Questions, Objectives, and Scope; Methodology, Methods, Tools, and Techniques. Research Ethics, Plagiarism, and Their Prevention.	8

Unit	Contents	Hours
Unit II	**Measurement, Data, and Analytics** Structured and Unstructured Data, Scales of Measurement, Population and Sample, Descriptive Statistics, Data Visualization. Probability and Random Variables, Sampling and Estimation, Hypothesis Testing, ANOVA, Correlation, and Regression. **Data Analytics:** Elements of Association, Clustering, and Classification.	8
Unit III	**Elements of Theoretical Research** Model and Model Building: Classification of Models, Exogenous and Endogenous Variables, Variable Relationships, Model Boundary, and Predictive and Prescriptive Models.	8
Unit IV	**Research Methods for Computer Science** **Formal Methods:** Formal Specification, Algorithm, and Complexity. **Building Artefacts:** Proof of Performance, Proof of Concept, and Proof of Existence. **Process Methodology:** Methods for Software Engineering and Human-Computer Interaction, Cognitive Processes, Interactive Games, Social Networks, and Web Analytics.	8
Unit V	**Research Documentation** Elements of Preparing a Paper and a Project report: Title, Abstract, Keywords, Acknowledgements, Symbols and Abbreviations, Introduction, Literature Review, Materials and Methods, Graphical and Tabular Presentation, Results and Discussion, Conclusion, Scope for Future Work, Citations and List of References, and Appendixes.	8

Text Books

Text Books
1. C. R. Kothari and G. Garg, *Research Methodology: Methods and Techniques*, Fourth Multi-Colour Edition, New Age International Publishers, 2019.

Course Outcomes
At the end of the course, the students should be able to: 1. Know and understand the basics and fundamentals of research problems, research methods, research work and its communication to the peers. 2. Understand the research methods specific to the field of computer science.

CSE602 – Compiler Design

Course Information	
Course Code	CSE602
Course Name	Compiler Design
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Theory of Computation

Course Objectives

1. To understand and list the different stages in the process of compilation.
2. Identify different methods of lexical analysis.
3. Design top-down and bottom-up parsers.
4. Identify synthesized and inherited attributes.
5. Develop syntax directed translation schemes.
6. Develop algorithms to generate code for a target machine.

Course Contents

Unit	Contents	Hours
Unit I	**Introduction to Compilers** Overview of the Translation Process, A Simple Compiler, Difference between interpreter, assembler and compiler. Overview and use of linker and loader, types of Compiler, Analysis of the Source Program, The Phases of a Compiler, Cousins of the Compiler, The Grouping of Phases, The Structure of a Compiler. Applications of Compiler Technology. **Lexical Analysis** The Role of the Lexical Analyzer, Specification and recognition of Tokens, The Lexical-Analyzer Generator Lex.	8
Unit II	**Syntax Analysis** Context-free language and grammar, push-down automata, LL(1) grammar and top-down parsing, operator grammar, LR(0), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (yacc, bison).	8
Unit III	**Syntax-Directed Translation** Attribute grammar, syntax directed definition, evaluation and flow of attribute in a syntax tree. **Symbol Table** Structure, Symbol attributes and management.	8

Unit	Contents	Hours
Unit IV	**Run-time Environment** Procedure activation, parameter passing, value return, memory allocation, and scope. **Intermediate Code Generation** Translation of different language features, different types of intermediate forms.	8
Unit V	**Code Improvement (Optimization)** Analysis: control-flow, data-flow dependence etc.; Code improvement local optimization, global optimization, loop optimization, peep-hole optimization etc. Register allocation and target code generation.	8

Text Books

Text Books
1. Alfred V. Aho, Ravi Sethi, Monica S. Lam and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, 2nd Ed., Pearson Education Ltd., 2007.
2. Alfred V. Aho and J. D. Ullman, *Principles of Compiler Design*, Narosa Publication, 2002.

References

References
1. Andrew W. Appel, *Modern Compiler Implementation in C/Java*, Cambridge University Press, 2003.
2. K. D. Cooper and L. Torczon, *Engineering a Compiler*, 2nd Ed., Morgan Kaufmann, 2011.

Course Outcomes
At the end of the course, the students should be able to: 1. For a given grammar specification develop the lexical analyser. 2. For a given parser specification design top-down and bottom-up parsers. 3. Develop syntax directed translation schemes. 4. Develop algorithms to generate code for a target machine.

CSE603 – Machine Learning

Course Information	
Course Code	CSE603
Course Name	Machine Learning
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Probability and Linear Algebra; Basic programming skills preferably in Python

Course Objectives

1. To impart the basic concepts of machine learning and algorithms.
2. To understand concepts about supervised and unsupervised learning.
3. To understand basic concepts about deep learning and learning theory.
4. To enable them to understand issues related to the application of machine learning algorithms.

Course Contents

Unit	Contents	Hours
Unit I	**Fundamentals** Introduction, Different Types of Learning, Hypothesis Space and Inductive Bias, Evaluation and Cross-Validation, Linear Regression, Introduction to Decision Trees, Learning Decision Tree, Overfitting.	8
Unit II	**Supervised Learning** Supervised learning setup, LMS, Logistic regression, Perceptron, Exponential family, Generative learning algorithms, Gaussian discriminant analysis, Naive Bayes, Support Vector Machines.	9
Unit III	**Learning Theory** Model selection and feature selection, Ensemble methods: Bagging, Boosting, Evaluating and debugging learning algorithms; Bias/variance tradeoff, Union and Chernoff/Hoeffding bounds, VC dimension, Worst-case (online) learning.	6
Unit IV	**Unsupervised Learning** Clustering: K-means, EM, Mixture of Gaussians, Factor Analysis, PCA (Principal Components Analysis), ICA (Independent Components Analysis).	6
Unit V	**Reinforcement Learning and Control** MDPs, Bellman equations, Value iteration and policy iteration, Linear Quadratic Regulation (LQR), LQG, Q-learning. Value function approximation, Policy search, Reinforce, POMDPs.	7

Text Books

Text Books
1. Ethem Alpaydin, *Introduction to Machine Learning*, 3rd Edition, PHI, 2015.
2. Tom M. Mitchell, *Machine Learning*, McGraw Hill Education, First Edition, 2017.

References

References
1. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition, 2009.
2. Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer, 2011 edition.

Course Outcomes

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

1. For a given problem, analyze and implement the solution using Linear Regression.
2. For a given problem, analyze and implement the solution using Logistic Regression, Decision Trees, and K-Nearest Neighbor.
3. Apply Bayesian learning and the Naive Bayes algorithm to appropriate learning problems.
4. Apply Support Vector Machines and kernels to solve classification problems.
5. Apply Neural Networks to determine and justify the correctness of a given solution.
6. Analyze hypothesis space, overfitting, bias and variance.
7. Analyze trade-offs between representational power and learnability.
8. Apply evaluation strategies and cross-validation and feature reduction methods.

CSE604 – Elective-I

Course Information	
Course Code	CSE604
Course Name	Elective-I (Name to be chosen from the list of elective courses offered by the Department of Computer Science and Engineering)
Category of Course	Professional Elective Courses (PE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Elective Course Provision
CSE604 represents the first Professional Elective course in the Semester VI curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular elective course selected by the student from the list of Professional Elective courses offered by the Department of Computer Science and Engineering.

CSE605 – Open Elective-I

Course Information	
Course Code	CSE605
Course Name	Open Elective-I (Name to be chosen from the list of approved Open Elective courses)
Category of Course	Open Elective Courses (OE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Open Elective Course Provision
CSE605 represents the Open Elective-I course in the Semester VI curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular Open Elective course selected by the student from the list of approved Open Elective courses.

CSE606 – Compiler Design Lab

Course Information	
Course Code	CSE606
Course Name	Compiler Design Lab
Category of Course	Professional Core Courses (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	NIL

Course Objectives
1. To understand and implement the principles, techniques, and also available tools used in compiler construction process. This will enable the students to work in the development phase of new computer languages in industry.

Experiments

1. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines.
2. Write a C program to identify whether a given line is a comment or not.
3. Write a C program to test whether a given identifier is valid or not.
4. Write a C program to simulate lexical analyzer for validating operators.
5. To study about Lexical Analyzer Generator (LEX) and Flex (Fast Lexical Analyzer).
6. Create a Lexer to take input from text file and count number of characters, number of lines, and number of words.
7. Design Predictive Parser for the given language.
8. Design a LALR bottom-up parser for the given language.
9. Convert the BNF rules into Yacc form and write code to generate abstract syntax tree.
10. Write a program to generate machine code from the abstract syntax tree generated by the parser.

Text Books / Reference Books

Text Books / Reference Books
1. Andrew W. Appel, *Modern Compiler Implementation in C*, Revised Edition, Cambridge University Press.
2. A. V. Aho and J. D. Ullman, *Principles of Compiler Design*, Pearson Education.
3. John R. Levine, Tony Mason, and Doug Brown, *Lex & Yacc*, O'Reilly.
4. Richard A. Loudon, *Compiler Construction*, Thomson.
5. K. D. Cooper and Linda Torczon, *Engineering a Compiler*, Elsevier.

6. Dick Grune, Henry E. Bal, Ceriel J. H. Jacobs, and Koen G. Langendoen, *Modern Compiler Design*, Wiley.

Course Outcomes

Upon the completion of Compiler Design practical course, the student will be able to:

1. Understand the working of Lex and Yacc compiler for debugging of programs.
2. Understand and define the role of lexical analyzer, use of regular expression and transition diagrams.
3. Understand and use Context Free Grammar and parse tree construction.
4. Learn and use the new tools and technologies used for designing a compiler.
5. Develop programs for solving parser problems.

CSE607 – Machine Learning Lab

Course Information	
Course Code	CSE607
Course Name	Machine Learning Lab
Category of Course	Professional Core Courses (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	CSE603 – Machine Learning; Basic Programming in Python

Course Objectives

The objectives of this laboratory course are:

1. To provide hands-on experience in implementing fundamental machine learning algorithms using Python and commonly used machine learning libraries.
2. To develop practical skills in data preprocessing, visualization, feature engineering, model development, and evaluation.
3. To implement and experimentally analyze supervised and unsupervised machine learning algorithms.
4. To provide practical exposure to model selection, cross-validation, feature selection, and performance evaluation.
5. To introduce students to popular machine learning frameworks and tools such as Scikit-learn, TensorFlow, and PyTorch.
6. To enable students to apply machine learning techniques to real-world datasets and interpret the results obtained from computational experiments.

Experiments

1. **Introduction to Python Libraries for Machine Learning**
   Study and implement basic operations using Python libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn. Load a dataset, inspect its structure, perform basic statistical analysis, and visualize selected features.
2. **Data Preprocessing and Visualization**
   Implement data preprocessing techniques including handling missing values, encoding categorical attributes, feature scaling, normalization, and train-test splitting. Visualize the data using suitable graphical representations.
3. **Linear Regression**
   Implement simple and multiple linear regression for predicting a continuous target variable. Evaluate the model using appropriate performance measures such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and coefficient of determination (R^2).
4. **Logistic Regression**
   Implement logistic regression for binary classification. Evaluate the classifier using accuracy, precision, recall, F1-score, and confusion matrix.
5. **Decision Tree Classification**
   Implement a decision tree classifier for a suitable dataset. Study the effect of tree depth and other parameters on model performance and demonstrate the issue of overfitting.

6. **K-Nearest Neighbour Classification**
   Implement the K-Nearest Neighbour (KNN) algorithm for classification. Experiment with different values of K and analyze their effect on classification performance.
7. **Naive Bayes Classification**
   Implement a Naive Bayes classifier for a suitable classification problem. Compare the performance of the classifier using different evaluation measures.
8. **Support Vector Machine**
   Implement Support Vector Machine (SVM) for classification. Experiment with different kernels such as linear, polynomial, and radial basis function (RBF), and compare their performance.
9. **Model Selection and Cross-Validation**
   Implement k-fold cross-validation for evaluating a machine learning model. Compare different models and hyperparameter settings using appropriate cross-validation strategies.
10. **Feature Selection and Dimensionality Reduction**
    Implement feature selection techniques and Principal Component Analysis (PCA) for dimensionality reduction. Compare model performance before and after feature reduction.
11. **Ensemble Learning**
    Implement ensemble learning techniques such as Bagging, Random Forest, and Boosting. Compare their performance with individual classification models.
12. **K-Means Clustering**
    Implement the K-Means clustering algorithm for an unlabeled dataset. Determine an appropriate number of clusters using suitable techniques and visualize the resulting clusters.

Text Books

Text Books
1. Ethem Alpaydin, *Introduction to Machine Learning*, 3rd Edition, PHI, 2015.
2. Tom M. Mitchell, *Machine Learning*, McGraw Hill Education, First Edition, 2017.

References

References
1. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition, Springer, 2009.
2. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
3. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd Edition, O'Reilly Media, 2022.

Course Outcomes

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

1. Implement fundamental machine learning algorithms using Python and appropriate machine learning libraries.
2. Perform data preprocessing, visualization, feature engineering, and dimensionality reduction for machine learning applications.
3. Implement and evaluate supervised learning algorithms including linear regression, logistic regression, decision trees, KNN, Naive Bayes, and Support Vector Machines.
4. Apply model selection, cross-validation, feature selection, and ensemble learning techniques to improve machine learning performance.
5. Implement unsupervised learning algorithms such as K-Means and hierarchical clustering and analyze the resulting clusters.
6. Use popular machine learning frameworks such as TensorFlow and PyTorch to develop and evaluate basic machine learning models.
7. Develop an end-to-end machine learning solution for a real-world problem and interpret the experimental results.

CSE608 – IT Workshop II (Mobile Application Development)

Course Information	
Course Code	CSE608
Course Name	IT Workshop II (Mobile Application Development)
Category of Course	Professional Core Courses (PC)
Credits	L:0 T:0 P:4 C:2
Pre-Requisites	Basic programming concepts; familiarity with object-oriented programming

Course Objectives

The objectives of this laboratory course are:

1. To introduce students to the fundamentals, architecture, and development lifecycle of mobile applications.
2. To develop practical skills in designing user interfaces and implementing event-driven mobile applications.
3. To provide hands-on experience in developing Android applications using Kotlin and Android Studio.
4. To introduce mobile application components, activities, services, intents, layouts, notifications, and device features.
5. To enable students to work with persistent storage, databases, RESTful APIs, JSON, and cloud-based services.
6. To introduce cross-platform mobile application development using Flutter/Dart.
7. To develop, test, debug, optimize, and deploy functional mobile applications.

Experiments

1. **Introduction to Mobile Application Development**
   Study the mobile application development ecosystem, mobile operating systems, native versus cross-platform applications, application lifecycle, mobile application architecture, and development workflow.
2. **Development Environment Setup**
   Install and configure Android Studio, Android SDK, emulator/device configuration, SDK tools, Gradle, and Kotlin development environment. Create, build, run, and debug a basic Android application.
3. **Version Control for Mobile Applications**
   Create a Git repository for a mobile application. Perform basic version-control operations including commit, branch, merge, and remote repository management.
4. **Android User Interface Development**
   Design mobile interfaces using layouts and UI components such as `TextView`, `EditText`, `Button`, `ImageView`, `CheckBox`, `RadioButton`, `Spinner`, and `RecyclerView`. Develop responsive interfaces suitable for different screen sizes.

5. **Event Handling and Activity Lifecycle**
   Develop an application using event-driven programming. Demonstrate activity lifecycle methods, event listeners, navigation between activities, and passing data between activities.
6. **Menus, Dialogues, Notifications and Material Components**
   Develop applications incorporating menus, dialogs, toast messages, snackbars, notifications, and commonly used Material Design components.
7. **Fragments and Navigation**
   Develop an application using fragments and implement navigation between screens using the Android navigation framework.
8. **Persistent Data Storage**
   Develop an application using local storage mechanisms such as SharedPreferences/DataStore and demonstrate storing and retrieving application data.
9. **SQLite/Room Database**
   Design and implement a mobile application using a local database. Perform basic CRUD operations using SQLite or the Room persistence library.
10. **Accessing Device Features**
    Develop applications that access selected device features such as camera, location/GPS, sensors, contacts, and Internet connectivity. Implement appropriate runtime permissions.
11. **Maps and Location-Based Services**
    Develop a location-aware application using maps and location services. Display the user's location and implement basic location-based functionality.
12. **REST API and JSON Processing**
    Develop a mobile application that consumes a RESTful web service. Retrieve JSON data from an API, parse the response, and display the information through the mobile interface.
13. **Integrated Mobile Application Project**
    Design and develop a complete mobile application addressing a real-world problem. The project should incorporate appropriate components such as user interface, navigation, local/cloud data storage, REST API or backend integration, authentication, device features where appropriate, error handling, testing, and documentation.

Text Books

Text Books
1. Neil Smyth, *Android Studio 4.0 Development Essentials: Developing Android 10 Apps Using Android Studio 4.0, Kotlin and Android Jetpack*, Payload Media.
2. Bill Phillips, Chris Stewart, Kristin Marsicano, and Brian Gardner, *Android Programming: The Big Nerd Ranch Guide*, Big Nerd Ranch.

References

References
1. Erik Hellman, *Android Programming: Pushing the Limits*, Wiley.
2. Barry A. Burd, *Android Application Development All-in-One for Dummies*, Wiley.
3. Marco L. Napoli, *Beginning Flutter: A Hands-On Guide to App Development*, Wiley.

4. Thomas Bailey and David Nagy, *Flutter Apprentice*, Kodeco.

Course Outcomes

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

1. Explain the architecture, components, lifecycle, and development workflow of mobile applications.
2. Design and implement responsive user interfaces for Android applications using Kotlin and Android Studio.
3. Develop mobile applications incorporating activities, fragments, navigation, event handling, notifications, and device-specific features.
4. Implement persistent data storage and database operations in mobile applications.
5. Develop mobile applications that communicate with RESTful APIs and process JSON-based data.
6. Integrate authentication, cloud databases, and backend services into mobile applications.
7. Apply testing, debugging, version control, packaging, and deployment practices in mobile application development.
8. Develop and demonstrate a functional mobile application addressing a real-world problem.

CSE609 – Minor Project

Course Information	
Course Code	CSE609
Course Name	Minor Project
Category of Course	Project (PR)
Credits	L:0 T:0 P:6 C:3
Pre-Requisites	Completion of the prescribed courses up to Semester V

General Description

The Minor Project is intended to provide students with an opportunity to apply the knowledge and skills acquired through the courses studied during the preceding semesters to a practical problem in Computer Science and Engineering. The project shall involve the identification of a suitable problem, formulation of objectives, study of relevant literature, requirement analysis, selection of appropriate methodologies and tools, design and development of a solution, experimentation, analysis of results, and technical documentation.

The project may be undertaken individually or in a small group, as permitted by the departmental guidelines. Students are expected to work under the supervision of a faculty member and demonstrate systematic planning, technical implementation, problem-solving ability, and effective technical communication.

The project may involve software development, algorithm design, data-driven applications, artificial intelligence and machine learning, computer networks, cybersecurity, databases, mobile or web applications, systems development, or other relevant areas of Computer Science and Engineering.

Project Objectives

The objectives of the Minor Project are to:

1. Enable students to identify and formulate a relevant problem in Computer Science and Engineering.
2. Apply theoretical knowledge and practical skills acquired from different courses to solve a real-world or research-oriented problem.
3. Develop skills in literature review, problem formulation, requirement analysis, system design, implementation, experimentation, and evaluation.
4. Provide experience in selecting and using appropriate programming languages, software tools, frameworks, datasets, and development methodologies.
5. Develop the ability to analyse experimental or computational results and draw meaningful conclusions.
6. Develop professional skills in project planning, teamwork, technical documentation, presentation, and communication.

Course Outcomes

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

1. Identify, formulate, and analyse a practical or research-oriented problem in Computer Science and Engineering.
2. Apply concepts, algorithms, methodologies, and tools acquired through the undergraduate curriculum to develop an appropriate solution.
3. Design, implement, test, and evaluate a software system, algorithm, model, prototype, or computational solution.
4. Analyse and interpret experimental or computational results using appropriate performance measures.
5. Demonstrate skills in project planning, teamwork, technical documentation, presentation, and professional communication.
6. Demonstrate the ability to critically evaluate the limitations of the developed solution and identify possible directions for future improvement.

DETAILED SYLLABUS
Fourth Year – 7th Semester

Sl.	Type	Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PC	CSE701	Cryptography and Network Security	3	0	0	3
2	PC	CSE702	Deep Learning	3	0	0	3
3	PE	CSE703(#)	Elective II	3	0	0	3
4	PE	CSE704(#)	Elective III	3	0	0	3
5	OE	CSE705(#)	Open Elective II	3	0	0	3
6	PR	CSE706	Internship (8 weeks)	0	0	0	2
7	PR	CSE707	Major Project I	0	0	8	4
						Total Credits	**21**

Note: # denotes the serial number of the elective, as listed under the Professional Electives or Open Electives, as applicable.

Note: Students selecting open electives offered by other departments are required to obtain the appropriate course codes directly from the respective departments offering those courses, to ensure accurate registration and documentation.

CSE701 – Cryptography & Network Security

Course Information	
Course Code	CSE701
Course Name	Cryptography & Network Security
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Computer Networks, Discrete Mathematics

Course Objectives

Students will try to learn:

1. The concepts of classical encryption techniques and concepts of finite fields and number theory.
2. The working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, and public key algorithms.
3. The design issues and working principles of various authentication protocols.
4. Various secure communication standards including Kerberos, IPsec, SSL/TLS, and secure email.
5. The ability to use existing cryptographic utilities to build programs for secure communication.
6. The concepts of cryptographic utilities and authentication mechanisms to design secure applications.

Syllabus

Unit	Contents
Unit I	**Introduction and Mathematical Foundations** Overview on Modern Cryptography, Ciphers and Secret Messages, CIA triad, Classical Cryptosystems, Cryptanalysis of Classical Cryptosystems, Digital Watermarking and Steganography. **Number Theory, Probability and Information Theory:** Mathematical Tools for Cryptography: Substitutions and Permutations, Modular Arithmetic, Euclid's Algorithm, Finite Fields, Polynomial Arithmetic, Discrete Logarithms.

Unit	Contents
Unit II	**Conventional and Modern Symmetric Encryption Algorithms** Theory of Block Cipher Design, Feistel Cipher Network Structures, DES and Triple DES, Modes of Operation (ECB, CBC, OFB, CFB). Modern Symmetric Encryption Algorithms: IDEA, Blowfish, AES-256, Authenticated Encryption, ChaCha20. Key Distribution. **Stream Ciphers and Pseudo Random Numbers:** Pseudo random sequences, Linear Congruential Generators, Cryptographic Generators, Design of Stream Cipher, One Time Pad, and Cryptanalysis of Symmetric Key Ciphers.
Unit III	**Public Key Cryptography, Hashes and Message Digests** Prime Numbers and Testing for Primality, Factoring Large Numbers, RSA, Diffie-Hellman, ElGamal, Key Exchange Algorithms, Public-Key Cryptography Standards. **Hashes and Message Digests:** Message Authentication, MD-5, SHA-(1,2,3), HMAC. Cryptanalysis of Asymmetric Key Ciphers, Modern Trends in Asymmetric Key Cryptography, Introduction to Post-Quantum Cryptography (PQC).
Unit IV	**Digital Signatures, Certificates, User Authentication** Digital Signature Standard (DSS and DSA), ECDSA, Security Handshake Pitfalls, Elliptic Curve Cryptosystems. **Authentication of Systems:** Kerberos V5, X.509 Authentication Service, OCSP.
Unit V	**Network Security** Secret Sharing Schemes, Network Protocols, Pretty Good Privacy (PGP), Transport Layer Security (TLS 1.3), Secure email, Zero Trust Security Architecture, Intruders and Viruses. **Intrusion Detection Systems:** Host-Based and Network-Based IDS, Honeypots. Firewalls, IPSec, Web Security, Access Control, Security in E-Commerce.

Text Books

Text Books
1. William Stallings, *Cryptography and Network Security: Principles and Practice*, 8th Edition, Pearson Education, 2023.
2. Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 20th Anniversary Edition, John Wiley & Sons, 2015.

References

References
1. Behrouz A. Forouzan, *Cryptography & Network Security*, Tata McGraw Hill.
2. Richard A. Mollin, *An Introduction to Cryptography*, 2nd Edition, CRC Press, 2006.
3. Jonathan Katz and Yehuda Lindell, *Introduction to Modern Cryptography*, 3rd Edition.

Course Outcomes

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

1. Identify information security goals, classical encryption techniques, and acquire fundamental knowledge on the concepts of finite fields and number theory.
2. Understand, compare, and apply different encryption and decryption techniques to solve problems related to confidentiality and authentication.
3. Apply the knowledge of cryptographic checksums and evaluate the performance of different message digest algorithms for verifying the integrity of varying message sizes.
4. Apply different digital signature algorithms to achieve authentication and create secure applications.
5. Apply network security basics, analyze different attacks on networks, and evaluate the performance of firewalls and security protocols such as SSL, IPSec, and PGP.
6. Apply the knowledge of cryptographic utilities and authentication mechanisms to design secure applications.

CSE702 – Deep Learning

Course Information	
Course Code	CSE702
Course Name	Deep Learning
Category of Course	Professional Core Courses (PC)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	Machine Learning / Soft Computing / Neural Networks

Course Objectives
The objectives of the course are: 1. To study deep learning and its applications in the field of computation. 2. To study the basics of neural networks and deep learning. 3. To study the concepts of gradient descent and Singular Value Decomposition. 4. To study various architectures of Convolutional Neural Networks. 5. To understand Recurrent Neural Networks and their architectures.

Syllabus

Unit	Contents
Unit I	**Fundamentals of Deep Learning** History of Deep Learning, Deep Learning Success Stories, McCulloch–Pitts Neuron, Perceptrons, Multilayer Perceptrons (MLPs), FeedForward Neural Networks, Backpropagation. Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam. Eigenvalues and eigenvectors. Principal Component Analysis (PCA) and its interpretations, Singular Value Decomposition.
Unit II	**Autoencoders and Regularization** Autoencoders and relation to PCA, Regularization in autoencoders, Denoising autoencoders, Sparse autoencoders, Contractive autoencoders. **Regularization:** Bias–Variance Tradeoff, L_2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout.
Unit III	**Deep Learning Optimization and Representation Learning** Greedy Layerwise Pre-training, Better activation functions, Better weight initialization methods, Batch Normalization. Learning Vectorial Representations of Words.

Unit	Contents
Unit IV	**Convolutional Neural Networks** Convolutional Neural Networks, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet. Visualizing Convolutional Neural Networks, Guided Backpropagation, Deep Dream, Deep Art, Fooling Convolutional Neural Networks.
Unit V	**Recurrent Neural Networks and Attention** Recurrent Neural Networks, Backpropagation through Time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, GRU, LSTMs. Encoder–Decoder Models, Attention Mechanism, Attention over Images.

Text Books

Text Books
1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press. http://www.deeplearningbook.org
2. LISA Lab, University of Montreal, *Deep Learning Tutorial*, 2015. http://deeplearning.net/tutorial/deeplearning.pdf

References

References
1. Li Deng and Dong Yu, *Deep Learning: Methods and Applications*. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/DeepLearning-NowPublishing-Vol7-SIG-039.pdf
2. Francois Chollet, *Deep Learning with Python*, Manning Publishing Co., 2018. https://tanthiamhuat.files.wordpress.com/2018/03/deeplearningwithpython.pdf

Course Outcomes
Upon successful completion of the course, students will be able to: 1. Interact with interdisciplinary course concepts. 2. Understand the concept of knowledge and knowledge base. 3. Demonstrate the skills of development of neural network-based intelligent systems for industrial problems. 4. Understand the design prerequisites and design procedure of intelligent systems. 5. Understand the concept of pattern classification and pattern association and implement them in project work.

CSE703 – Elective-II

Course Information	
Course Code	CSE703
Course Name	Elective-II (Name to be chosen from the list of elective courses offered by the Department of Computer Science and Engineering)
Category of Course	Professional Elective Courses (PE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Elective Course Provision
CSE703 represents the second Professional Elective course in the Semester VII curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular elective course selected by the student from the list of Professional Elective courses offered by the Department of Computer Science and Engineering.

CSE704 – Elective-III

Course Information	
Course Code	CSE704
Course Name	Elective-III (Name to be chosen from the list of elective courses offered by the Department of Computer Science and Engineering)
Category of Course	Professional Elective Courses (PE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Elective Course Provision
CSE704 represents the third Professional Elective course in the undergraduate curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular elective course selected by the student from the list of Professional Elective courses offered by the Department of Computer Science and Engineering.

CSE705 – Open Elective-II

Course Information	
Course Code	CSE705
Course Name	Open Elective-II (Name to be chosen from the list of approved Open Elective courses)
Category of Course	Open Elective Courses (OE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Open Elective Course Provision
CSE705 represents the Open Elective-II course in the Semester VII curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular Open Elective course selected by the student from the list of approved Open Elective courses.

CSE706 – Internship (8 Weeks)

Course Information	
Course Code	CSE706
Course Name	Internship (8 Weeks)
Category of Course	Project / Internship (PR)
Credits	L:0 T:0 P:0 C:2
Pre-Requisites	Completion of the prescribed courses up to Semester VI
Duration	8 Weeks

General Description
The eight-week internship is intended to provide students with practical exposure to professional practices, industrial environments, research organizations, academic institutions, or technology-oriented projects. The internship shall enable students to apply the knowledge and skills acquired during the undergraduate programme to real-world problems and professional activities. During the internship, students are expected to work under the guidance of an industry professional, research supervisor, faculty member, or other designated mentor. The internship may involve software development, algorithm design, data science, artificial intelligence and machine learning, cybersecurity, computer networks, databases, cloud computing, web or mobile application development, systems development, research activities, or other relevant areas of Computer Science and Engineering. The internship shall provide students with an opportunity to understand professional work culture, project planning, teamwork, technical communication, problem-solving, documentation, quality practices, and ethical responsibilities associated with the computing profession.

Internship Objectives

The objectives of the internship are to:

1. Provide students with practical exposure to professional work environments and real-world applications of Computer Science and Engineering.
2. Enable students to apply theoretical knowledge and technical skills acquired through the undergraduate curriculum to practical problems.
3. Develop professional competencies in problem-solving, programming, system design, analysis, testing, experimentation, and documentation.
4. Familiarize students with industry-standard tools, technologies, development methodologies, and professional practices.
5. Develop skills in teamwork, communication, project planning, time management, and technical reporting.
6. Enable students to understand professional ethics, workplace responsibilities, intellectual property, data security, and responsible use of technology.
7. Encourage students to explore emerging technologies and identify potential areas for further study, research, or professional development.

Course Outcomes

Upon successful completion of the eight-week internship, students will be able to:

1. Apply the knowledge and technical skills acquired through the undergraduate curriculum to real-world problems in Computer Science and Engineering.
2. Analyse practical problems, identify appropriate requirements, and select suitable technologies, tools, algorithms, and methodologies.
3. Design, implement, test, and evaluate a software system, algorithm, model, prototype, experiment, or other technical solution as applicable to the internship.
4. Demonstrate professional skills including teamwork, communication, time management, documentation, and project planning.
5. Apply appropriate professional ethics, responsible computing practices, data security principles, and workplace responsibilities.
6. Analyse and communicate technical results through reports, presentations, demonstrations, and other professional forms of communication.
7. Identify opportunities for further learning, research, innovation, or professional development based on the internship experience.

CSE707 – Major Project I

Course Information	
Course Code	CSE707
Course Name	Major Project I
Category of Course	Project (PR)
Credits	L:0 T:0 P:8 C:4
Pre-Requisites	Completion of the prescribed courses up to Semester VI

General Description

Major Project I constitutes the first phase of the major project to be continued in the subsequent semester. The course is intended to provide students with an opportunity to identify and investigate a significant problem in Computer Science and Engineering and to develop a systematic solution using appropriate engineering, computational, analytical, or research methodologies.

The project may be undertaken individually or in a small group, as permitted by the Department, under the supervision of a faculty member. Students are expected to identify a relevant problem, conduct a systematic literature survey, formulate the problem and objectives, analyse requirements, identify suitable methodologies and technologies, design the proposed system or solution, and undertake preliminary implementation, experimentation, or validation.

Major Project I should establish the technical foundation for the continuation and completion of the project in Major Project II. The work completed during this phase should result in a clearly defined problem, well-justified methodology, preliminary results or prototype, and a concrete plan for the subsequent phase of the project.

Project Objectives

The objectives of Major Project I are to:

1. Enable students to identify and formulate a significant problem in Computer Science and Engineering.
2. Develop the ability to conduct a systematic literature survey and critically analyse existing approaches related to the selected problem.
3. Enable students to define clear project objectives, scope, requirements, constraints, and expected outcomes.
4. Develop skills in selecting appropriate algorithms, methodologies, architectures, tools, frameworks, datasets, and technologies.
5. Enable students to design an appropriate system, model, algorithm, prototype, or experimental framework for the selected problem.
6. Provide practical experience in implementing or experimentally validating the proposed approach.
7. Develop skills in analysing preliminary results and identifying limitations and possible improvements.
8. Prepare a technically sound foundation and work plan for the successful completion of the project in Major Project II.

Suggested Project Activities

Major Project I may be carried out through the following stages:

1. **Problem Identification and Selection:** Identify a relevant engineering, computational, or research problem and define its scope and significance.
2. **Literature Review:** Study relevant research papers, books, standards, technical reports, datasets, software tools, and existing solutions. Identify research gaps, limitations, and opportunities for improvement.
3. **Problem Formulation:** Formulate the problem clearly and define the objectives, research questions, assumptions, constraints, and expected outcomes.
4. **Requirement Analysis:** Identify functional and non-functional requirements, data requirements, hardware and software requirements, and other project constraints.
5. **Methodology Selection:** Select and justify suitable algorithms, models, architectures, experimental methods, development methodologies, tools, and technologies.
6. **System Design / Model Development:** Develop the system architecture, workflow, algorithms, database design, experimental framework, mathematical model, or other appropriate design artefacts.
7. **Initial Implementation / Experimentation:** Develop the initial prototype, implement the proposed algorithms or system components, prepare datasets, conduct simulations, or perform preliminary experiments as applicable.
8. **Testing and Preliminary Evaluation:** Perform initial testing and evaluation using appropriate datasets, benchmarks, performance measures, or experimental criteria.
9. **Analysis of Results:** Analyse the preliminary results, identify limitations, compare with appropriate baseline or existing approaches, and determine areas for improvement.
10. **Planning for Major Project II:** Prepare a detailed plan for completing implementation, experimentation, evaluation, documentation, and final validation in Major Project II.

DETAILED SYLLABUS
Fourth Year – 8th Semester

Sl.	Type	Code	Course Title	Hours/Week			Credits
				L	T	P	
1	PE	CSE801(#)	Elective IV	3	0	0	3
2	PE	CSE802(#)	Elective V	3	0	0	3
3	OE	CSE803(#)	Open Elective III	3	0	0	3
4	PR	CSE804	Grand Viva	0	0	2	1
5	PR	CSE805	Major Project II	0	0	12	6
Total Credits							**16**

CSE801 – Elective-IV

Course Information	
Course Code	CSE801
Course Name	Elective-IV (Name to be chosen from the list of elective courses offered by the Department of Computer Science and Engineering)
Category of Course	Professional Elective Courses (PE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Elective Course Provision
CSE801 represents the fourth Professional Elective course in the undergraduate curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular elective course selected by the student from the list of Professional Elective courses offered by the Department of Computer Science and Engineering.

CSE802 – Elective-V

Course Information	
Course Code	CSE802
Course Name	Elective-V (Name to be chosen from the list of elective courses offered by the Department of Computer Science and Engineering)
Category of Course	Professional Elective Courses (PE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Elective Course Provision
CSE802 represents the fifth Professional Elective course in the undergraduate curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular elective course selected by the student from the list of Professional Elective courses offered by the Department of Computer Science and Engineering.

CSE803 – Open Elective-III

Course Information	
Course Code	CSE803
Course Name	Open Elective-III (Name to be chosen from the list of approved Open Elective courses)
Category of Course	Open Elective Courses (OE)
Credits	L:3 T:0 P:0 C:3
Pre-Requisites	As applicable to the selected elective course

Open Elective Course Provision
CSE803 represents the Open Elective-III course in the Semester VIII curriculum. The course title, prerequisites, course objectives, syllabus, text books, references, and course outcomes shall be as specified for the particular Open Elective course selected by the student from the list of approved Open Elective courses.

CSE804 – Grand Viva

Course Information	
Course Code	CSE804
Course Name	Grand Viva
Category of Course	Project / Viva (PR)
Credits	L:0 T:0 P:2 C:1
Pre-Requisites	Successful completion of the prescribed courses up to Semester VII

Objectives
The objectives of the Grand Viva are to:
1. Assess the overall knowledge, understanding, and application of concepts acquired during the undergraduate programme.
2. Evaluate the student's ability to think logically, critically, creatively, and independently.
3. Assess communication skills, confidence, clarity of expression, listening ability, and the ability to participate constructively in a discussion.
4. Evaluate awareness and informed opinion regarding contemporary national and international affairs.
5. Assess awareness of developments in science, technology, computing, artificial intelligence, society, environment, economy, and other areas relevant to modern life.
6. Understand the student's interests, hobbies, extracurricular activities, achievements, and personal areas of engagement.
7. Assess awareness and appreciation of sports, games, literature, arts, culture, and other dimensions of social and cultural life.
8. Evaluate leadership qualities, teamwork, interpersonal skills, adaptability, initiative, and sense of responsibility.
9. Assess ethical reasoning, social awareness, professional values, and responsible citizenship.
10. Understand the student's career aspirations, future plans, and preparedness for employment, higher education, research, entrepreneurship, or other professional pursuits.

Scope of the Grand Viva

The Grand Viva may cover, but need not be limited to, the following broad areas:

1. **Academic and Technical Knowledge:** Fundamental and advanced concepts of Computer Science and Engineering, programming, algorithms, data structures, databases, operating systems, computer networks, software engineering, artificial intelligence, machine learning, cybersecurity, and other relevant areas studied during the programme.
2. **Major Project and Internship:** Problem statement, objectives, methodology, implementation, experimentation, results, learning experiences, challenges, and future scope of the student's major project and internship.
3. **Problem-Solving and Critical Thinking:** Logical reasoning, analytical thinking, estimation, interpretation of situations, decision-making, and approaches to solving unfamiliar problems.
4. **Current Affairs and General Awareness:** Awareness of significant national and international developments, including developments in science and technology, economy, education, environment, society, geopolitics, and other important contemporary issues.
5. **Science, Technology and Innovation:** Emerging technologies, artificial intelligence, robotics, quantum computing, cybersecurity, biotechnology, space technology, sustainable technology, digital transformation, and other contemporary innovations.
6. **Social and Environmental Awareness:** Social responsibilities, sustainable development, climate and environmental concerns, digital inclusion, ethical use of technology, privacy, responsible artificial intelligence, and the social impact of technological developments.
7. **Sports and Games:** Interest in sports and games, knowledge of major sporting events, sportspersons, teams, achievements, rules, and the student's own participation or experiences

in sports and physical activities.

8. **Hobbies and Personal Interests:** Discussion of the student's hobbies, reading habits, music, travel, photography, writing, programming, creative activities, collecting, volunteering, or other areas of genuine personal interest.
9. **Arts, Literature and Culture:** Awareness and appreciation of literature, music, cinema, visual arts, performing arts, heritage, traditions, and cultural diversity.
10. **Leadership and Teamwork:** Experiences in student organizations, clubs, events, projects, volunteering, community activities, leadership roles, teamwork, and conflict resolution.
11. **Ethics and Values:** Professional ethics, academic integrity, responsible use of technology, respect for diversity, social responsibility, personal values, and ethical decision-making.
12. **Communication and Personality:** Ability to articulate ideas clearly, engage in discussion, listen carefully, respond appropriately to questions, defend an opinion with reasoning, and communicate respectfully with others.
13. **Career and Future Aspirations:** Career goals, higher education, research interests, entrepreneurship, professional aspirations, personal development plans, and awareness of the skills required for future professional growth.

CSE805 – Major Project II

Course Information	
Course Code	CSE805
Course Name	Major Project II
Category of Course	Project (PR)
Credits	L:0 T:0 P:12 C:6
Pre-Requisites	Successful completion of Major Project I (CSE707)

Objectives

The objectives of Major Project II are to:

1. Continue and extend the project work initiated during Major Project I.
2. Implement the proposed methodology, architecture, algorithms, models, or system developed during the initial phase of the project.
3. Develop, integrate, test, and refine the proposed solution or prototype.
4. Conduct systematic experiments, simulations, case studies, or performance evaluations appropriate to the project.
5. Analyse and interpret the results obtained from the implementation or experimentation.
6. Compare the proposed solution with appropriate existing methods, systems, algorithms, or benchmarks wherever applicable.
7. Identify the strengths, limitations, challenges, and possible sources of error associated with the developed solution.
8. Develop the ability to work independently and systematically on an open-ended engineering problem.
9. Develop technical documentation and communicate the project work effectively through a comprehensive project report and presentation.
10. Demonstrate the ability to apply knowledge acquired during the undergraduate programme to solve a significant Computer Science and Engineering problem.

Major Project II Activities

The project work may include, as appropriate to the nature of the project:

- Detailed implementation of the proposed system or methodology.
- Algorithm or model development and refinement.
- Software development and system integration.
- Dataset preparation, data collection, preprocessing, and analysis.
- Simulation and experimental investigations.
- Development of prototypes or proof-of-concept systems.
- Testing and debugging.
- Performance evaluation and benchmarking.
- Statistical analysis and interpretation of results.
- Comparison with existing approaches.

- Optimization and refinement of the developed solution.
- Documentation of technical design, implementation, experiments, and results.
- Preparation and presentation of the final project outcomes.

Course Outcomes

Upon successful completion of Major Project II, students will be able to:

1. Extend and complete a substantial Computer Science and Engineering project initiated during Major Project I.
2. Apply appropriate concepts, algorithms, methodologies, tools, and technologies to develop a solution to a significant engineering problem.
3. Implement, integrate, test, and validate a software system, model, algorithm, framework, prototype, or experimental solution as appropriate to the project.
4. Design and conduct appropriate experiments, simulations, case studies, or performance evaluations.
5. Analyse, interpret, and critically evaluate project results using appropriate qualitative and quantitative measures.
6. Compare the developed solution with existing approaches or benchmarks and identify its strengths and limitations.
7. Demonstrate independent learning, systematic problem-solving, project management, and technical decision-making abilities.
8. Communicate technical work effectively through professional documentation, presentation, and demonstration.
9. Demonstrate the ability to integrate knowledge acquired from different areas of Computer Science and Engineering in solving a significant real-world or research-oriented problem.
10. Demonstrate professional competence and readiness to undertake advanced technical, research, or industry-oriented work.

ELECTIVE COURSES

Course Syllabi

Professional Electives and Open Electives

Course-wise syllabus, objectives, references and outcomes

PROFESSIONAL ELECTIVES

Advanced Learning and Professional Specialization

Advanced Algorithms

Course Objectives

- Introduce students to the advanced methods of designing and analysis of algorithms in computing.
- Students should be able to select appropriate algorithm techniques for a specific problem.
- Students learn about the different modeling of problem-solving like data structures, graph, decomposing the problem, which is used to solve the advanced algorithmic issues.
- Students should be able to classify the different classes of problems based on their computational difficulties.

Syllabus

UNIT I

Overview of Design Paradigms

Background: Motivation, the role of algorithms in computing, Analyzing of algorithms, algorithms like heap sort, search algorithms, etc. Designing techniques: overview of Divide and Conquer, Greedy method, Dynamic Programming, Branch and Bound, Backtracking, Graph traversal algorithms.

UNIT II

Matroids, String and Graph Matching

Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST. String Matching: Introduction to string-matching problem, Naïve algorithm, Rabin Karp, String matching with finite automata, Knuth-Morris-Pratt algorithms and complexity analysis. Graph Matching: Algorithm to compute maximum matching.

Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path.

UNIT III

Max-Flow Problem and Matrix Computation

Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. Matrix Computations: Strassen's algorithm and introduction to divide and conquer

paradigm, inverse of a triangular matrix, relation between the time complexities

of basic matrix operations, LUP-decomposition.

UNIT IV

Shortest Path Problems, Modulo Representation of integers/polynomials, Discrete Fourier Transform (DFT):

Shortest Path Problems in Graphs: Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming. Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo representation. Discrete Fourier Transform: DFT and FFT algorithms.

UNIT V

Theory of NP-Hard and NP-Completeness Problems, and Approximation Algorithms:

Theory of NP-Hard and NP-Completeness Problems: P, NP, and NP-Complete complexity classes, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs. Approximation Algorithms: Notion of NP-completeness: P class, NP-hard class, NP-complete class, Circuit Satisfiability problem, Clique Decision Problem,etc. Approximation Algorithms: Approximation algorithms for vertex-cover problem, set cover, TSP, knapsack, subset-sum problem etc.

Text Books

Text Books

1. Introduction to Algorithms, 3rd Edition, T H. Cormen, C E. Leiserson, R L. Rivest, and Clifford Stein, published by PHI Learning Private Limited (Original edition published by the MIT Press, Cambridge, MA, USA), Copyright © 2011.
2. The Design and Analysis of Computer Algorithms by Aho, Hopcroft, Ullma.

Reference Books

Reference Books

1. Algorithm Design, 1st Edition, Jon Kleinberg, and Eva Tardos, published by Pearson Education Limited, Copyright © 2014.
2. Algorithms, 1st Edition, S. Dasgupta, C. Papadimitriou, and U. Vazirani, published by McGraw-Hill Education, Copyright © 2008.

Course Outcomes

- Students will be able to apply the concept and design strategies to algorithm design.
- Also, capable of writing the correctness of algorithms systematically.
- Students will be able to choose a suitable data structure for solving the problems, and also design the appropriate algorithms.
- Students will be capable of classifying the different class of problems based on their completeness theorem.
- Students will be able to understand more details in the field of advanced data structures for synthesizing more complicated problems in the field of engineering.

Advanced Computer Architecture

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Advanced Computer Architecture
Category of Course	Professional Elective Courses
Credits	3L:0T: 0P C:3
Pre-Requisites	Computer Organization and Architecture

Course Objectives

The student should be made to:

- Understand the micro-architectural design of processors.
- Learn about the various techniques used to obtain performance improvement and power savings in current processors.

Syllabus

UNIT I

Pipelining Basic And Intermediate Concept

Review of Fundamentals of CPU, Memory and IO, Trends in technology, power, energy and cost, Dependability, Performance Evaluation,Review of Pipelining, Examples of some pipeline in modern processors, pipeline hazards, data hazards, and control hazards. Techniques to handle hazards, performance improvement with pipelines and effect of hazards on the performance.

UNIT II

Instruction Level Parallelism

ILP concepts, Pipelining overview, Compiler Techniques for Exposing ILP,

Dynamic Branch Prediction, Dynamic Scheduling, Multiple instruction Issue, Hardware Based Speculation, Static scheduling, Multi-threading, Limitations of ILP, Case Studies of Intel core i7 and ARM Cortex A8.

UNIT III

Data-Level Parallelism Vector architecture, SIMD parallelism.

extensions,

Graphics

Processing

Units,

Loop

level

UNIT IV

Thread Level Parallelism

Symmetric and Distributed Shared Memory Architectures, Performance Issues, Synchronization, Models of Memory Consistency.

Case studies

Intel i7 Processor, SMT & CMP Processors.

UNIT V

Cache Performance

Reducing Cache Miss Penalty and Miss Rate, Reducing Hit Time, Main Memory and Performance, Memory Technology.

Text Books

Text Books

1. John L Hennessey and David A Patterson, “Computer Architecture A Quantitative Approach”, Morgan Kaufmann/ Elsevier, Fifth Edition, 2012.

Reference Books

Reference Books

1. Kai Hwang and Faye Briggs, “Computer Architecture and Parallel Processing”, McGraw-Hill International Edition, 2000.
2. Sima D, Fountain T and Kacsuk P, ”Advanced Computer Architectures: A Design Space Approach”, Addison Wesley, 2000.

Course Outcomes

- At the end of the course, the student should be able to:
- Evaluate performance of different architectures with respect to various parameters.
- Study about different hazards and its resolution.
- Analyze performance of different ILP techniques.
- Identify cache and memory related issues in multi-processors.

Advance Database Management Systems

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Advance Database Management Systems
Credits	3L:0T: 0P (3 Credits)
Pre-Requisites	Database Management System

Course Objectives

- Master transaction processing, concurrency control and crash recovery
- Master query processing and optimization
- Master advanced indexing and data organization for DBMS
- Master different types of Databases

Syllabus

UNIT I

Introduction to different types of databases, Normal Forms and advanced normal forms, Introduction to Relational Algebra, SQL and PL- SQL

UNIT II

Transaction Concepts, Transaction states, Concurrent Execution, Concurrency Control like Lock based protocols, Timestamp – Based protocols etc, Recovery Systems.

UNIT III

Indexing and Hashing, B – Tree, B+ - Tree, ISAM, R – Trees, R*– Trees etc

UNIT IV

Distributed Databases, Distributed storage, Distributed Transactions, Commit protocols, Concurrency Control in Distributed Databases, Distributed Query Processing

UNIT V

Introduction to Parallel Databases, I/O parallelism, Interquery parallelism, Inraquery Parallelism, Interoperation and intraoperation parallelism.

Text Books

Text Books

1. Database System Concepts, International Edition, Silberscatz, Korth, Sudarshan.

2. Fundamentals of Database Systems, 6th Edition, Elmasri and Navathe

Course Outcomes

- Explain in detail DBMS architecture.
- Explain in detail advanced query processing and techniques involved in query optimization.
- Explain the principles of concurrency control.
- Explain the principles of recovery management.

Advanced Operating System

Course Objectives

- To Understand the working of a distributed Operating system.
- To understand the issues in designing a distributed Operating System.
- To understand the synchronization primitives of interaction of distributed Operating System.
- To understand the construct and functioning of Distributed shared – memory and Deadlock management in distributed environment.
- To understand the various failure modes of the system and failure recovery in a distributed environment.

Syllabus

UNIT I

Distributed Systems

Architectures of Distributed Systems, System Architecture types, Issues in distributed operating systems, Communication networks, Communication primitives. Concept of a Process, Concurrent Processes, The Critical Section Problem, Other Synchronization Problems, Language Mechanisms for Synchronization, Axiomatic Verification of Parallel Programs.

UNIT II

Theoretical Foundations

Inherent limitations of a distributed system, Lamport's logical clocks, Vector clocks, Causal ordering of messages, Global state, Cuts of a distributed computation, Termination detection. Distributed Mutual Exclusion, Introduction, The classification of mutual exclusion and associated algorithms, A comparative performance analysis.

UNIT III

Distributed Deadlock Detection

Introduction, Deadlock handling strategies in distributed systems, Issues in deadlock detection and resolution, Control organizations for distributed deadlock detection, Centralized and distributed deadlock detection algorithms, Hierarchical deadlock detection algorithms. Agreement protocols, Introduction-the system model, a classification of agreement problems, Solutions to the Byzantine agreement problem, Applications of agreement algorithms.

Distributed resource management

Introduction, Architecture, Mechanism for building distributed file

systems, Design issues, Log structured file systems.

UNIT IV

Distributed shared memory

Architecture, Algorithms for implementing DSM, Memory coherence and protocols, Design issues. Distributed Scheduling, Introduction, Issues in load distributing, Components of a load

distributing algorithm, Stability, Load distributing algorithm, Performance comparison, Selecting a suitable load sharing algorithm, Requirements for load distributing, Task migration and associated issues.

UNIT V

Failure Recovery and Fault tolerance

Introduction, Basic concepts, Classification of failures, Backward and Forward error recovery, Backward error recovery, Recovery in concurrent systems, Consistent set of check.

Check Points

Synchronous and asynchronous checkpointing and recovery, Checkpointing for distributed database systems, Recovery in replicated distributed databases.

Text Books

Text Books

1. Mukesh Singhal, Niranjan G. Shivaratri, ”Advanced concepts in operating systems: Distributed, Database and multiprocessor operating systems”, TMH, 2001

Reference Books

Reference Books

1. Andrew S. Tanenbaum, ”Modern operating system”, PHI, 2003
2. Pradeep K.Sinha, ”Distributed operating system-Concepts and design”, PHI, 2003.Andrew S.Tanenbaum, ”Distributed operating system”, Pearson education, 2003

Course Outcomes

- The course will help the students to understand the basic aim and scope of Distributed Operating System.
- The Course will help students in analyzing the various issues in designing a Distributed Operating system and also give insight into various solutions to overcome the issues at hand.
- Deep understanding of deadlock handling and synchronization primitives of various algorithms in distributed environment.
- Understanding of process scheduling and implementation of memory coherence, load balancing, processor to processor interaction.
- Deep understanding of Failure recovery and fault tolerance.

Advanced Software Development

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Advanced Software Development
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Software Engineering, Object-Oriented Programming, Data Structures and Fundamentals of Artificial Intelligence

Course Objectives

The course aims to:

- Introduce the fundamental concepts and architectures of AI-agent-based software systems.
- Develop an understanding of Large Language Models, prompting, tool use, retrieval, memory, planning, and reasoning in agentic systems.
- Enable students to design and develop basic AI agents and agent-based software applications.
- Introduce the fundamental concepts of multi-agent systems and agent collaboration.
- Develop awareness of the evaluation, security, reliability, and responsible use of AI agents.

Syllabus

UNIT I

Introduction to AI Agents and Agentic Software

Evolution of software development towards intelligent software systems; Artificial Intelligence and Large Language Models in software development; Large Language Model capabilities and limitations; introduction to AI agents; characteristics of intelligent and autonomous agents; conventional software versus agentic software; agent perception, reasoning, planning and action; basic agent architectures; reactive and goal-oriented agents; agent interaction with software environments.

UNIT II

Fundamentals of Agent Architecture

Components of an AI agent; agent control loop; prompts and context; prompt engineering; structured outputs; reasoning and planning; task decomposition; function calling and tool use; external APIs and services; agent state; short-term and long-term memory; basic reflection and self-correction; human-in-the-loop agents; introduction to agent development frameworks.

UNIT III

Knowledge, Retrieval and Tool-Augmented Agents

Knowledge augmentation for AI agents; Retrieval-Augmented Generation (RAG); document processing and semantic search; embeddings and vector databases; retrieval and context construction; tool selection and tool execution; database and API integration; grounding agent responses using external knowledge; handling tool failures; basic agent workflows; applications of tool-augmented agents.

UNIT IV

Fundamentals of Multi-Agent Systems

Introduction to multi-agent systems; single-agent versus multi-agent architectures; agent roles and specialization; task delegation; communication between agents; coordination and collaboration; planner and worker agents; supervisor-based architectures; shared state and memory; basic conflict resolution; human–agent and agent–agent interaction; applications of multi-agent systems.

UNIT V

Development, Evaluation and Safety of AI Agents

Basic development lifecycle of an AI agent; designing agent workflows; testing agent behaviour; evaluation of task completion and response quality; reliability and robustness; hallucination and error handling; monitoring and logging agent behaviour; prompt injection and unsafe tool use; data privacy and access control; guardrails and human oversight; responsible development and deployment of AI-agent-based software; emerging applications of AI agents in software development.

Text Books

Text Books

1. Chip Huyen, *AI Engineering: Building Applications with Foundation Models*, O'Reilly Media.
2. Sebastian Raschka, *Build a Large Language Model (From Scratch)*, Manning Publications.

Reference Books

Reference Books

1. Martin Kleppmann, *Designing Data-Intensive Applications*, O'Reilly Media.
2. Sam Newman, *Building Microservices*, 2nd Edition, O'Reilly Media.
3. Martin Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley.

Course Outcomes

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

- Explain the fundamental concepts, characteristics, and architectures of AI-agent-based software systems.
- Design basic AI agents using Large Language Models, prompting, planning, memory, and tool-use mechanisms.
- Develop simple retrieval- and tool-augmented agents using external knowledge sources, APIs, and databases.
- Explain the fundamental concepts of multi-agent systems and design basic agent collaboration and task-delegation workflows.
- Evaluate basic AI-agent behaviour and identify issues related to reliability, security, privacy, safety, and responsible deployment.

Artificial Intelligence

Course Objectives

- To introduce the foundational concepts, history, and scope of Artificial Intelligence.
- To develop the ability to formulate problems as search problems and apply uninformed, informed, and adversarial search strategies.
- To study knowledge representation techniques and logic-based reasoning for building intelligent systems.
- To introduce planning and reasoning under uncertainty using probabilistic methods.
- To understand the design and development of rule-based/expert systems and their real-world applications.
- To create awareness of the ethical, societal, and safety considerations associated with AI systems.

Syllabus

UNIT I

- Introduction to AI and Intelligent Agents

Foundations and history of AI, the Turing Test, scope and applications of AI, AI vs Machine Learning

vs Expert Systems.

Intelligent agents – agent and environment, rationality, PEAS (Performance, Environment, Actuators,

Sensors) description, types of environments, structure of agents – simple reflex, model-based, goal-

based, utility-based, and learning agents.

UNIT II

- Problem Solving by Search

Problem formulation, state space representation.

Uninformed search – breadth-first search, depth-first search, uniform-cost search, iterative deepening.

Informed (heuristic) search – greedy best-first search, A* search, heuristic design and admissibility,

local search methods – hill climbing, simulated annealing.

Adversarial search – game playing, the minimax algorithm, alpha-beta pruning, basics of Monte Carlo

Tree Search.

UNIT III

- Knowledge Representation and Reasoning

Knowledge-based agents, propositional logic – syntax, semantics, inference, resolution.

First-order logic – syntax, semantics, quantifiers, unification, and inference; forward and backward

chaining.

Structured knowledge representation – semantic networks, frames, ontologies; rule-based systems.

UNIT IV

- Planning and Reasoning under Uncertainty

Classical planning – representation of planning problems, state-space and plan-space planning, partial-

order planning (overview).

Sources of uncertainty in AI; probability theory refresher – Bayes' theorem, conditional

independence.

Probabilistic reasoning – Bayesian networks, construction and inference; brief introduction to Markov

Decision Processes as a bridge to reinforcement learning.

UNIT V

- Expert Systems and AI Applications

Expert systems – architecture, knowledge engineering, human vs artificial experts, algorithmic vs

heuristic methods, forward and backward chaining in rule-based expert systems.

Expert system development – tools and shells, knowledge base editors, explanation facilities,

development phases, reliability and maintainability, case examples.

Overview of AI application areas – natural language processing, computer vision, robotics,

recommendation systems (as covered in depth in respective electives).

Ethical and societal issues in AI – bias and fairness, transparency and explainability, safety,

accountability, and responsible AI development.

Text Books

Text Books

1. 1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson.
2. 2. Elaine Rich, Kevin Knight, Shivashankar B. Nair, "Artificial Intelligence", 3rd Edition, McGraw Hill.

Reference Books

Reference Books

1. 1. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, Pearson Education.
2. 2. Peter Jackson, “Introduction to Expert Systems”, 3rd Edition, Pearson Education.
3. 3. Donald A. Waterman, “A Guide to Expert Systems”, Pearson.
4. 4. Nils J. Nilsson, “Artificial Intelligence: A New Synthesis”, Morgan Kaufmann.

Course Outcomes

- On completion of this course, students will be able to:
- Explain the foundational concepts of AI and describe intelligent agents in terms of PEAS and agent architectures.
- Formulate real-world problems as search problems and apply appropriate uninformed, informed, or adversarial search techniques.
- Represent knowledge using propositional and first-order logic and perform logical inference.
- Apply planning techniques and probabilistic reasoning (including Bayesian networks) to problems involving uncertainty.
- Design a simple rule-based/expert system for a given problem domain.
- Critically evaluate the ethical, societal, and safety implications of deploying AI systems.

Bioinformatics

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Bioinformatics
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Data Structures, Algorithms and Basic Biology

Course Objectives

The course aims to:

- Introduce the fundamental concepts of bioinformatics and computational biology.
- Develop an understanding of biological data, biological databases, and sequence representation.
- Introduce fundamental computational techniques for DNA, RNA and protein sequence analysis.
- Familiarize students with sequence alignment, similarity analysis, phylogenetic analysis, and basic genome analysis.
- Introduce the applications of bioinformatics in modern biological research, healthcare, and drug discovery.

Syllabus

UNIT I

Introduction to Bioinformatics and Biological Data

Introduction to bioinformatics and computational biology; scope and applications of bioinformatics; biological information and computational approaches; DNA, RNA and proteins; genes and genomes; genetic code; biological sequence representation; sequence formats; biological data types; primary and secondary biological databases; introduction to NCBI, GenBank, EMBL and UniProt; basic concepts of data retrieval and biological database searching.

UNIT II

Biological Sequence Analysis

Sequence similarity and sequence homology; sequence comparison; pairwise sequence alignment; global and local alignment; scoring matrices; gap penalties; dynamic programming; Needleman–Wunsch algorithm; Smith–Waterman algorithm; sequence similarity searching; introduction to BLAST; interpretation of alignment results.

UNIT III

Multiple Sequence and Phylogenetic Analysis

Multiple sequence alignment; motivation and applications; basic approaches for multiple sequence alignment; conserved regions and motifs; sequence patterns; introduction to phylogenetics; evolutionary relationships; distance methods; phylogenetic trees; tree representation and interpretation; applications of sequence alignment and phylogenetic analysis.

UNIT IV

Genome and Protein Analysis

Fundamentals of genome analysis; genome organization; gene prediction; open reading frames; promoter and regulatory regions; comparative genomics; basic concepts of transcriptomics; protein sequence and structure; amino acid properties; protein motifs and domains; protein sequence databases; introduction to protein structure prediction and functional annotation.

UNIT V

Applications and Computational Tools in Bioinformatics

Introduction to structural bioinformatics; molecular docking and virtual screening; basics of drug discovery using computational methods; biomarker identification; disease and genomic data analysis; introduction to biological data visualization; bioinformatics workflows and reproducible analysis; introduction to commonly used bioinformatics tools and online resources; applications of bioinformatics in medicine, agriculture, biotechnology, and drug discovery.

Text Books

Text Books

1. Teresa K. Attwood and David J. Parry-Smith, *Introduction to Bioinformatics*, Pearson Education.
2. Arthur M. Lesk, *Introduction to Bioinformatics*, Oxford University Press.

Reference Books

Reference Books

1. David W. Mount, *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press.
2. Richard Durbin, Sean R. Eddy, Anders Krogh and Graeme Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press.
3. Jonathan Pevsner, *Bioinformatics and Functional Genomics*, Wiley-Blackwell.

Course Outcomes

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

- Explain the fundamental concepts of bioinformatics, biological sequences, genomes, proteins, and biological databases.
- Retrieve and analyse biological sequence information using appropriate databases and bioin-

formatics resources.

- Apply fundamental sequence alignment and similarity-search techniques to DNA, RNA, and protein sequences.
- Interpret multiple sequence alignments, conserved regions, motifs, and basic phylogenetic relationships.
- Explain the applications of bioinformatics in genomics, protein analysis, healthcare, biotechnology, and drug discovery.

Blockchain

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Blockchain
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Data Structures, Computer Networks and Basic Cryptography

Course Objectives

The course aims to:

- Introduce the fundamental concepts, principles, and architecture of blockchain technology.
- Develop an understanding of distributed ledgers, cryptographic techniques, transactions, blocks, and consensus mechanisms.
- Explain the operation and characteristics of public, private, and consortium blockchain networks.
- Introduce smart contracts and decentralized applications based on blockchain platforms.
- Familiarize students with the applications, benefits, limitations, security issues, and emerging use cases of blockchain technology.

Syllabus

UNIT I

Introduction to Blockchain Technology

Introduction to blockchain and distributed ledger technology; evolution of blockchain; characteristics of blockchain systems; centralized, decentralized and distributed systems; blockchain architecture; blocks, transactions and chains; distributed ledgers; peer-to-peer networks; immutability and transparency; blockchain nodes; wallets and digital assets; types of blockchain networks; public, private and consortium blockchains; overview of major blockchain platforms.

UNIT II

Cryptography and Blockchain Fundamentals

Role of cryptography in blockchain; cryptographic hash functions; properties of secure hash functions; SHA family; digital signatures; public and private keys; asymmetric cryptography; transaction authentication; Merkle trees; cryptographic integrity; addresses and wallets; basic concepts of cryptocurrency and tokenization.

UNIT III

Consensus and Blockchain Networks

Need for consensus in distributed systems; Byzantine Generals Problem; consensus mechanisms; Proof of Work; Proof of Stake; delegated and alternative consensus approaches; mining and block validation; transaction verification; fork and chain selection; network propagation; incentives and rewards; basic concepts of blockchain scalability; limitations of consensus mechanisms.

UNIT IV

Smart Contracts and Decentralized Applications

Introduction to smart contracts; characteristics and lifecycle of smart contracts; smart contract platforms; Ethereum and Ethereum Virtual Machine (EVM); accounts and transactions; gas and transaction fees; introduction to Solidity; basic smart contract structure; contract deployment and execution; tokens and token standards; decentralized applications (DApps); interaction between smart contracts and users; basic security considerations in smart contracts.

UNIT V

Blockchain Applications, Security and Challenges

Applications of blockchain in finance, supply chain management, healthcare, identity management, education, and digital assets; decentralized identity; asset and document management; blockchain-based voting; security threats and attacks; 51% attack; double spending; Sybil attacks; smart contract vulnerabilities; privacy issues; scalability and interoperability challenges; energy and environmental considerations; regulatory and governance issues; limitations and future directions of blockchain technology.

Text Books

Text Books

1. Melanie Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly Media.
2. Andreas M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, 2nd Edition, O'Reilly Media.

Reference Books

Reference Books

1. Andreas M. Antonopoulos and Gavin Wood, *Mastering Ethereum: Building Smart Contracts and DApps*, O'Reilly Media.
2. Imran Bashir, *Mastering Blockchain*, 4th Edition, Packt Publishing.
3. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, *Bitcoin and Cryptocurrency Technologies*, Princeton University Press.

Course Outcomes

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

- Explain the fundamental concepts, architecture, and characteristics of blockchain and distributed ledger technologies.
- Apply basic cryptographic concepts used for securing blockchain transactions and maintaining data integrity.
- Explain the principles of blockchain consensus mechanisms and transaction validation.
- Develop basic smart contracts and explain the architecture of decentralized applications.
- Analyse the major applications, security issues, limitations, and challenges associated with blockchain technology.

Cloud Computing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Cloud Computing
Category of Course	Professional Elective Courses
Credits	3L:0T: 0P C:3
Pre-Requisites	Operating Systems, Virtualization Technologies, Networking.

Course Objectives

- The student will also learn how to apply trust-based security model to real-world security problems.
- An overview of the concepts, processes, and best practices needed to successfully secure information within Cloud infrastructures.
- Students will learn the basic Cloud types and delivery models and develop an understanding of the risk and compliance responsibilities and Challenges for each Cloud type and service delivery model.

Syllabus

UNIT I

Introduction to Cloud Computing, The Evolution of Cloud Computing, Hardware Evolution, Internet Software Evolution, Server Virtualization, Web Services Deliver from the Cloud, Communication-as-a-Service, Infrastructure-as-a-Service, Monitoring-as-a-Service, Platform-as-a-Service, Software-as-a-Service, Building Cloud Network.

UNIT II

Federation in the Cloud, Presence in the Cloud, Privacy and its Relation to Cloud-Based Information Systems, Security in the Cloud, Common Standards in the Cloud, End User Access to the Cloud Computing.

UNIT III

Introduction, Advancing towards a Utility Model, Evolving IT infrastructure, Evolving Software Applications, Continuum of Utilities, Standards and Working Groups, Standards Bodies and Working Groups, Service Oriented Architecture, Business Process Execution Language, Interoperability Standards for Data Center Management,

Utility Computing Technology, Virtualization, Hyper Threading, Blade Servers, Automated Provisioning, Policy Based Automation, Application Management, Evaluating Utility Management Technology, Virtual Test and development Environment, Data Center Challenges and Solutions, Automating the Data Center.

UNIT IV

Software Utility Application Architecture, Characteristics of an SaaS, Software Utility Applications, Cost Versus Value, Software Application Services Framework, Common

Enablers, Conceptual view to Reality, Business Profits, Implementing Database Systems for Multitenant Architecture.

UNIT V

Other Design Considerations, Design of a Web Services Metering Interface, Application Monitoring Implementation, A Design for an Update and Notification Policy, Transforming to Software as a Service, Application Transformation Program, Business Model Scenarios, Virtual Services for Organizations, The Future.

Text Books

Text Books

1. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice), Tim Mather, ISBN-10: 0596802765,O'Reilly Media, September 2009.
2. Bunker and Darren Thomson, "Delivering Utility Computing", 2006, John Wiley & Sons Ltd.

Reference Books

Reference Books

1. John W. Rittinghouse and james F. Ransome, "Cloud Computing Implementation, Management and Security", 2010, CRC Press, Taylor & Francis Group, Boca Raton London New York. [Unit -11 and Unit II].
2. Alfredo Mendoza, "Utility Computing Technologies, Standards, and Strategies", Artech House INC, 2007. [Unit -11I to Unit V] 3.George Reese, "Cloud Application Architectures", O'Reilly Publications, 2009.

Course Outcomes

- After completion of course, students would be able to:
- Identify security aspects of each cloud model.
- Develop a risk-management strategy for moving to the Cloud.
- Implement a public cloud instance using a public cloud service provider.
- Apply trust-based security model to different layer.

Computational Complexity

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Computational Complexity
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Data Structure, Algorithm, Theory of Computation

Course Objectives

The Student should be made to:

- Learn the main computational complexity classes, their underlying models of computation, and relationships.
- Understand the concept of reductions and its role in classifying problems by their computational complexity.
- Be able to show using reductions that a problem is NP-complete.
- Be familiar with the concepts of randomized, approximation, and parallel algorithms and aware of the related complexity classes.

Syllabus

UNIT I

Introduction

Easy and hard problems. Algorithms and complexity. Turing machines: Models of computation. Multi-tape deterministic and non-deterministic Turing machines. Decision problems.

UNIT II

The Halting Problem and Undecidable Languages

Counting and diagonalization. Tape reduction. Universal Turing machine. Undecidability of halting. Reductions. Rice's theorem. Deterministic Complexity Classes: DTIME[t]. Linear Speed-up Theorem. P Time. Polynomial reducibility. Polytime algorithms: 2-satisfiability, 2-colourability.

UNIT III

NP and NP-completeness

Non-deterministic Turing machines. NTIME[t]. NP. Polynomial time verification. NPcompleteness. Cook-Levin Theorem. Polynomial transformations: 3- satisfiability, clique, colourability, Hamilton cycle, partition problems. Pseudo-polynomial time. Strong NP-completeness. Knapsack. NP-hardness.

UNIT IV

Space Complexity and Hierarchy Theorems

Space complexity and hierarchy theorems. DSPACE[s]. Linear Space Compression Theorem. PSPACE, NPSPACE. PSPACE = NPSPACE. PSPACE-completeness. Quantified Boolean Formula problem is PSPACE-complete. L, NL and NL-completeness. NL = coNL. Hierarchy theorems.

UNIT V

Randomized Complexity, Interactive Proof Systems and Optimization

The classes BPP, RP, ZPP. Interactive proof systems: IP = PSPACE.

Optimization and approximation: Combinatorial optimization problems. Relative error. Bin-packing problem. Polynomial and fully polynomial approximation schemes. Vertex cover, traveling salesman problem, minimum partition.

Computational Geometry

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Computational Geometry
Category of Course	Professional Elective Courses
Credits	3L:0T: 0P C:3
Pre-Requisites	Operating Systems

Course Objectives

- To provide an account of fundamental concepts of quantitative geometry and
- To discuss graphical techniques of geometric construction with experiments using computers.

Syllabus

UNIT I

Polygon Triangulation

Polygons-Jordan Curve Theorem-The Art Gallery Theorem-Fisk's Proof of Sufficiency, Triangulation: theory-Existence of a Diagonal-Properties of Triangulations-Triangulation Dual-3-Coloring Proof. Area of polygon and its theorem.

Polygon Partitioning

Monotone partitioning, Trapezoidalization, Monotone Mountains. Convex partitioning.

UNIT II

Convex Hulls in Two Dimensions

Definitions of convexity and convex hulls,Extreme points and Naive algorithms for extreme points-Extreme Edges QuickHull,Graham's Algorithm, Lower Bound, Incremental Algorithm, Divide and Conquer Polyhedra, Regular Polytopes-Euler's Formula. Hull Algorithms-incremental algorithm and complexity, Polyhedral Boundary Representations.

UNIT III

Voronoi Diagrams

Applications: Preview, Definitions and Basic Properties, Halfplanes, Size of Diagram, Delaunay Triangulations, Properties of Delaunay Triangulations, Properties of Voronoi Diagrams, Algorithms,Applications in Detail-Nearest Neighbors, Largest Empty CircleMinimum Spanning Tree-Traveling Salesperson Problem.

UNIT IV

Arrangements

Voronoi Diagrams & Medial Axis, Connection to Convex Hulls, Connection to Arrangements, Combinatorics of Arrangements, Combinatorics of Arrangements, Incremental Algorithm.

UNIT V

Duality, Higher-Order Voronoi Diagrams, Applications, Segment-Segment Intersection, Segment-Triangle Intersection.

Text Books

Text Books

1. Computational Geometry in C 2nd edition by Joseph O'Rourke(Cambridge university press).

Reference Books

Reference Books

1. Computational Geometry Algorithms and Applications Third Edition, Springer by Mark de Berg · Otfried Cheong Marc van Kreveld · Mark Overmars.

Course Outcomes

- At the end of the course, the student should be able to:
- explain the basic principles and theory of geometric algorithms,
- learn and improve their algorithmic skills
- apply the techniques to specific application domains of interest
- develop their own algorithms for solving geometric problems.

Computer Graphics

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Computer Graphics
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Analytic Geometry, Linear Algebra, Basic Programming

Course Objectives

- Gain knowledge about graphics hardware devices and software used.
- Understand the two-dimensional graphics and their transformations.
- Understand the three-dimensional graphics and their transformations.
- Appreciate illumination and color models.
- Be familiar with understand clipping techniques. Provide an understanding of mapping from a world coordinates to device coordinates, clipping, and projections.
- To be able to discuss the application of computer graphics concepts in the development of computer games, information visualization, and business applications.
- To comprehend and analyze the fundamentals of animation, virtual reality, underlying technologies, principles, and applications.

Syllabus

UNIT I

Introduction to Computer Graphics & Graphics Systems

Overview of computer graphics & its uses, Classification, characteristics & components & applications of computer graphics, Representing pictures, Basic Terminologies: Pixel, Resolution & its different types, Dots, Dot Pitch, Aspect ratio; Raster scan display: Refresh rate & Interlacing, Retrace, CRT, Shadow mask, Aperture grill, Bit planes, Color depth, Color palette, Frame buffer, Video controller, General architecture of Raster Scan display, Active & Passive graphics devices, Computer graphics software.

UNIT II

Scan Conversion

Points & lines, Line drawing algorithms: DDA algorithm, Advantages & Disadvantages; Bresenham's line algorithm; Circle generation algorithm: Basic concepts, DDA circle drawing algorithm, Midpoint circle drawing algorithm, Bresenham's circle drawing algorithm; Ellipse generation algorithm: Basic concepts, Midpoint ellipse generation algorithm; Aliasing, Antialiasing, Methods of antialiasing.

UNIT III

Polygon Filling algorithms and Transformation Inside & Outside test of polygon: Even-Odd method, Winding number method; Polygon filling algorithms: Scan line polygon, Scan line seed fill algorithm, Boundary fill algorithm, Flood fill algorithm. Basic transformations Translation, rotation, scaling, reflection, shear; Transformation between coordinate systems; Homogeneous coordinates & Combined transformations; Inverse transformation: Rotation about an arbitrary point, General fixed-point scaling, Reflection through an arbitrary line.

UNIT IV

Viewing and Clipping

Viewing transformation, Viewing pipeline, Window to viewport co-ordinate transformation; Clipping: Point clipping, Line clipping – Cohen-Sutherland algorithm, Liang-Barsky algorithm, Polygon clipping – Sutherland-Hodgeman algorithm, Weiler–Atherton algorithm; Text clipping.

3D transformation and Projection:

3D transformations

Translation, rotation, scaling, reflection & shearing. Rotation about an axis parallel to a coordinate axis, Rotation about an arbitrary axis in space, reflection through an arbitrary plane; 3D Projection: Parallel projection – Orthographic, Axonometric, Oblique; Perspective projection – transformation matrix, vanishing points, Single-point, Two-point, Three-point perspective transformation.

UNIT V

Curves, Hidden Surfaces, Color and Shading Models

Curve generation algorithm- DDA method, approximation method, Spline representation, Continuity, Piecewise Cubic Spline, Bezier curves – Cubic Bezier, Mid-point Bezier, Bspline curves; Depth comparison, Z-buffer algorithm, Back faces detection, BSP tree method, Painter's algorithm, Scan-line algorithm; Hidden line elimination method, wire frame methods; Introduction, Modeling Light Intensities and Sources, Diffuse Reflection, Lambert's Cosine Law, Specular Reflection, Half-toning, Color Models - RGB Color, CMY Color.

Text Books

Text Books

1. Computer Graphics Multimedia and Animation, Malay K. Pakhira, 2nd Ed., PHI Learning Pvt. Ltd.
2. D. P. Mukherjee, Fundamentals of Computer Graphics & Multimedia, Prentice Hall.
3. D. F. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, McGraw Hill.

Reference Books

Reference Books

1. D. Hearn and M. P. Baker, Computer Graphics (C version), Prentice Hall.
2. J. F. K Buford., Multimedia Systems, Pearson Education.

3. P. K. Andleigh and K. Thakrar, Multimedia Systems Design, Pearson Education India
4. S. Harrington, Computer Graphics: A programming Approach, McGraw Hill.
5. V. Dam; F. H. John; J. D. Foley; S. K. Feiner, Computer Graphics principles and practice, Pearson Education.
6. W. M. Newman and R. F. Sproull, Principles of Interactive computer Graphics, McGraw Hill.
7. M. E. Cook, Principles of Interactive Multimedia, McGraw Hill.
8. Mukhopadhyay and A. Chattopadhyay, Introduction to Computer Graphics and Multimedia, Vikas Publishing House.

Course Outcomes

- Design two-dimensional graphics.
- Apply two dimensional transformations.
- Design three-dimensional graphics.
- Apply three dimensional transformations.
- Apply Illumination and color models.
- Apply clipping techniques to graphics.
- Design animation sequences.

Computer Vision

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Computer Vision
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Image Processing

Course Objectives

The students will be able to:

- Develop a fundamental understanding of Computer Vision concepts and techniques.
- Apply Computer Vision algorithms to solve real-world image and video processing problems.
- Explore advanced topics in Computer Vision, including deep learning-based approaches.
- Analyze and critique state-of-the-art research in Computer Vision.

Syllabus

UNIT I

Introduction to Computer Vision

Overview of Computer Vision and its significance, Image formation and representation (pixels, color models), Image processing fundamentals (filters, histograms, transforms), Image enhancement techniques (contrast adjustment, noise reduction), Introduction to Computer Vision libraries and tools (OpenCV).

UNIT II

Feature Detection and Matching

Feature extraction and keypoint detection (Harris corner detection, SIFT, SURF), Image descriptors and matching techniques (ORB, BRISK, BRIEF), Object detection and recognition (Haar cascades, template matching), Optical character recognition (OCR) using Tesseract, Hands-on exercises on feature-based image matching.

UNIT III

Image Segmentation and Object Tracking

Image segmentation algorithms (thresholding, region-based, edge-based), Motion estimation and object tracking methods (Lucas-Kanade, Mean-Shift), Video analysis and tracking applications (surveillance, motion detection), Case studies in object tracking using OpenCV, Project: Real-time object tracking and analysis.

UNIT IV

Deep Learning for Computer Vision

Introduction to deep learning and neural networks, Convolutional Neural Networks (CNNs) for image analysis, Transfer learning and pre-trained models (VGG, ResNet), Object detection with CNNs (YOLO, Faster R-CNN), Hands-on exercises with deep learning frameworks (TensorFlow, PyTorch).

UNIT V

Advanced Topics in Computer Vision

3D computer vision and depth perception, Face recognition and biometric systems, Image-based medical diagnosis and applications (MRI, CT), Emerging trends in Computer Vision research (GANs, generative models).

Text Books

Text Books

1. Szeliski R., “Computer Vision: Algorithms and Applications”, Springer, 2010.
2. Introductory techniques for 3-D computer vision, Trucco, Emanuele & Verri, Alessandro. published by Pearson (1998), ISBN – 0132611082.
3. Concise Computer Vision: An Introduction into Theory and Algorithms, Dr. Reinhard Klette, published by Springer Nature; 2014 edition, ISBN: 1447163192.

Reference Books

Reference Books

1. Shapiro L. G. and Stockman G., “Computer Vision”, Prentice Hall, 2001.
2. Forsyth D. A. and Ponce J., “Computer Vision – A Modern Approach”, Second Edition, Pearson Education, 2012.
3. Davies E. R., “Machine Vision: Theory, Algorithms, Practicalities”, Morgan Kaufmann, 2004.

Course Outcomes

- Students will be able to apply fundamental Computer Vision concepts to solve image and video processing challenges effectively.
- Students will be capable of implementing Computer Vision algorithms for tasks such as feature detection, object recognition, and motion tracking.
- Students will be able to utilize deep learning techniques in Computer Vision to develop advanced applications.
- Students will be able to evaluate and critique state-of-the-art research papers and stay current with developments in the field.
- Students will be able to demonstrate practical competence by successfully applying Computer Vision knowledge to real-world projects and applications.

Cybersecurity

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Cybersecurity
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Cryptography and Network Security

Course Objectives

The students will be able to:

- Understand the fundamentals of cyberspace and cybersecurity terminology.
- Identify common cyber threats and attack vectors.
- Implement cybersecurity best practices and data protection measures.
- Comprehend the legal and ethical aspects of cybercrime and cybersecurity.
- Analyze real-world cybersecurity incidents and case studies.

Syllabus

UNIT I

Basic Introduction to Cyberspace and Cybersecurity

Introduction to cyberspace and its significance in the modern world. Key terminology in cybersecurity (e.g., malware, firewall, penetration testing). Evolution of cybersecurity from its early days to current trends. Roles and responsibilities of cybersecurity professionals in organizations. Cybersecurity ethics and professional conduct.

UNIT II

Cybercrime and Cybersecurity Threat Landscape

Understanding different types of cyber threats (viruses, worms, ransomware, DDoS attacks). Common cyber-attack vectors (e.g., social engineering, phishing, malware distribution). Analysing the global threat landscape and recent cyber incidents. Case studies of prominent cyber attacks (e.g., WannaCry, Equifax breach). The economics of cybercrime and its impact on businesses and individuals.

UNIT III

Solutions or Precautions to Resist Cyber Threats and Data Privacy

Cybersecurity best practices for individuals and organizations. Developing information security policies and procedures. Security awareness and training programs. Data privacy principles and

regulations (e.g., GDPR, HIPAA). Data encryption and protection measures (e.g., SSL, encryption algorithms).

UNIT IV

Cybercrime and Cyber Law

Cybercrime laws and regulations at the national and international levels. Cybercrime and the legal landscape around the world. IT Act, 2000 and its amendments. Limitations of the IT Act, 2000. Investigating cybercrimes: digital forensics and evidence collection. Case studies of high-profile cybercrime prosecutions. Ethical hacking and penetration testing as legal and ethical practices.

UNIT V

Emerging Trends in Cybersecurity

Internet of Things (IoT) security challenges and solutions. Cloud security and virtualization best practices. Artificial intelligence and machine learning in cybersecurity (e.g., anomaly detection). Blockchain and cryptocurrency security considerations. Biometric security technologies and their use in authentication.

Text Books

Text Books

1. William Stallings, “Cryptography and Network Security: Principles and Practices”, 6th Edition, Pearson Education Ltd., 2016.
2. Karnika Seth, “Computers, Internet and New Technology Laws”, Cyber Lawyer and Expert and Managing Partner of Seth Associates.
3. S. K. Verma and Raman Mittal, “Legal Dimensions of Cyber Space”, Indian Law Institute, New Delhi, 2004.
4. Thomas Halt, Adam M. Bossler and Kathryn C. Seigfried-Spellar, “Cybercrime and Digital Forensics: An Introduction”, Routledge, Taylor and Francis Group, 2017.

Reference Books

Reference Books

1. Sumit Belapure and Nina Godbole, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives”, Wiley India Pvt. Ltd.
2. Dorothy F. Denning, “Information Warfare and Security”, Addison Wesley.
3. Henry A. Oliver, “Security in the Digital Age: Social Media Security Threats and Vulnerabilities”, CreateSpace Independent Publishing Platform.

Course Outcomes

- Students will be able to apply foundational concepts and terminology in cybersecurity to analyze and address security challenges effectively.
- Students will be able to evaluate and assess various cyber threats and vulnerabilities systematically, demonstrating the ability to recognize and mitigate security risks.

- Students will be able to design and implement cybersecurity solutions and strategies to protect data and information systems.
- Students will be able to comprehend the legal and ethical dimensions of cybersecurity, demonstrating an understanding of relevant laws and regulations.
- Students will be able to analyze and investigate real-world cyber incidents and case studies, drawing meaningful insights to enhance security measures.

Data Analytics

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Data Analytics
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Data Structure, Statistics, Data Mining

Course Objectives

The Student should be made to:

- Be exposed to big data.
- Learn the different ways of Data Analysis.
- Be familiar with data streams.
- Learn the mining and clustering.
- Be familiar with the visualization.

Syllabus

UNIT I

Introduction to Big data

Introduction to Big Data Platform – Challenges of conventional systems - Web data – Evolution of Analytic scalability, analytic processes and tools, Analysis vs. reporting - Modern data analytic tools,

Statistical concepts

Sampling distributions, resampling, statistical inference, and prediction error.

UNIT II

Data analysis

Regression modeling, Multivariate analysis, Bayesian modeling, inference and Bayesian networks, and Support vector and kernel methods, Analysis of time series: linear systems analysis, nonlinear dynamics - Rule induction - Neural networks: learning and generalization, competitive learning, principal component analysis and neural networks.

Fuzzy logic

Extracting fuzzy models from data, fuzzy decision trees, Stochastic search methods.

UNIT III

Mining data streams: Introduction to Streams Concepts – Stream data model and architecture - Stream Computing, Sampling data in a stream – Filtering streams – Counting distinct elements in a stream – Estimating moments – Counting oneness in a window – Decaying window - Real-time Analytics Platform (RTAP) applications - case studies - real time sentiment analysis, stock market predictions.

UNIT IV

Frequent item sets and clustering Mining Frequent item sets - Market based model – Apriori Algorithm – Handling large data sets in Main memory – Limited Pass algorithm – Counting frequent item sets in a stream – Clustering Techniques – Hierarchical – K- Means – Clustering high dimensional data – CLIQUE and PROCLUS – Frequent pattern based clustering methods – Clustering in non-Euclidean space – Clustering for streams and Parallelism.

UNIT V

Frameworks and visualization

MapReduce – Hadoop, Hive, MapR – Sharding – NoSQL Databases - S3

- Hadoop Distributed file systems – Visualizations - Visual data analysis techniques, interaction techniques; Systems and applications.

Text Books

Text Books

1. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, ”Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today’s Businesses
2. Montgomery, Douglas C. and Runger, George C. (2014) Applied Statistics and Probability for Engineers, 6th edition, John Wiley & Sons, Inc (ISBN- 978-1118539712) Reference books:
3. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
4. Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 2012.
5. P. J. Sadalage and M. Fowler, ”NoSQL Distilled: A Brief Guide to the Emerging World of
6. Tom White, ”Hadoop: The Definitive Guide”, Third Edition, O’Reilly, 2012.
7. Eric Sammer, ”Hadoop Operations”, O’Reilly, 2012.

Course Outcomes

- After completion of course, students would be:
- Demonstrate understanding of hypotheses testing for samples to solve engineering problems.
- Perform linear and multiple linear regression analyses.
- Demonstrate ability to design and analysis of single-factor experiments.
- Demonstrate ability to do design of experiments with several factors.

- Describe big data and use cases from selected business domains.
- Explain NoSQL big data management.
- Install, configure, and run Hadoop and HDFS.
- Perform map-reduce analytics using Hadoop.
- Use Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics.

Data Mining

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Data Mining
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic Programming Skills General Understanding Of Statistics

Course Objectives

- To introduce data warehousing and mining techniques.
- Be acquainted with the tools and techniques used for Knowledge Discovery in Databases.

Syllabus

UNIT I

Introduction to Data Warehousing; Data Mining: Mining frequent patterns, association and correlations; Sequential Pattern Mining concepts, primitives, scalable methods; Integration of a Data Mining System with a Data Warehouse; Data Preprocessing.

UNIT II

Mining Frequent Patterns, Associations and Correlations; Mining Methods; Mining various Kinds of Association Rules; Correlation Analysis, Constraint Based Association Mining, Classification and Prediction, Basic Concepts, Decision Tree Induction; Bayesian Classification; Rule Based Classification; Classification by Back propagation, Support Vector Machines, Associative Classification; Lazy Learners; Other Classification Methods; Prediction; Cluster Analysis – Types of Data in Cluster Analysis.

UNIT III

Cluster Analysis; Types of Data; Categorization of Major Clustering Methods, K-means, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid Based Methods, Model-Based Clustering Methods; Clustering High Dimensional Data, Constraint, Based Cluster Analysis, Outlier Analysis, Data Mining Applications.

UNIT IV

Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, and Similarity search in Time-series analysis.

UNIT V

Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Classification of dynamic data streams, Class Imbalance Problem; Graph

Mining; Social Network Analysis;

Text Books

Text Books

1. Jiawei Han, M Kamber and J Pei " Data Mining Concepts and Techniques", Third Edition, Elsevier Publication, 2011.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar "Introduction to Data Mining", Pearson Education, 2007

Reference Books

Reference Books

1. Alex Berson and Stephen J. Smith "Data Warehousing, Data Mining & OLAP", Tata McGraw – Hill Edition, Tenth Reprint 2007.
2. G.K. Gupta – Introduction to Data Mining with case Studies, PHI, New Delhi – 2006.
3. A. Berson& S.J. Smith – Data Warehousing Data Mining, COLAP, TMH, New Delhi,2004

Course Outcomes

- After completion of course, students would be:
- Apply data mining techniques and methods to large data sets.
- Use data mining tools.
- Compare and contrast the various classifiers.

Digital Forensics

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Digital Forensics
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3

Course Objectives

The students will be able to:

- Understand the principles and methodologies of digital forensics for investigating cybercrimes and incidents.
- Develop proficiency in acquiring and analyzing digital evidence from various devices and sources.
- Apply forensic tools and techniques to recover, preserve, and present digital evidence effectively.
- Explore legal and ethical considerations in digital forensics, including chain of custody and expert witness testimony.

Syllabus

UNIT I

Introduction to Digital Forensics

Introduction to digital forensics and its role in cybercrime investigation. Legal and ethical considerations in digital investigations (chain of custody, expert witness ethics). Overview of the digital forensics process and methodologies. Types of digital evidence and sources (computers, mobile devices, cloud, network logs).

UNIT II

Digital Evidence Acquisition

Data acquisition methods and strategies (disk imaging, memory capture). Best practices in preserving and securing evidence. Mobile device forensics techniques (smartphones, tablets). Network forensics: capturing and analyzing network traffic.

UNIT III

Digital Evidence Analysis

Data recovery techniques (file system reconstruction, deleted file recovery). File system analysis (NTFS, FAT, ext4, HFS+). Registry analysis and artifact extraction (Windows Registry). Carving data from unallocated space and fragmented files.

UNIT IV

Forensic Tools and Techniques

Introduction to digital forensic software and hardware tools (EnCase, FTK, Autopsy). Data validation and verification methods (hashing, checksums). Timeline analysis and reconstruction of digital events. Detecting and analyzing steganography and encryption in digital media.

UNIT V

Case Studies and Practical Applications

Real-world digital forensic investigations and case studies. Analyzing cybercrime cases and incident response scenarios. Preparing for expert witness testimony in court. Emerging trends and challenges in digital forensics (IoT forensics, cloud forensics).

Text Books

Text Books

1. Warren G. Kruse II and Jay G. Heiser, “Computer Forensics: Incident Response Essentials”, Addison-Wesley, 2002.
2. Nelson, B., Phillips, A., Enfinger, F., and Stuart, C., “Guide to Computer Forensics and Investigations”, 2nd ed., Thomson Course Technology, 2006, ISBN: 0-619-21706-5.

Reference Books

Reference Books

1. Vacca, J., “Computer Forensics: Computer Crime Scene Investigation”, 2nd ed., Charles River Media, 2005, ISBN: 1-58450-389.

Course Outcomes

- Students will be able to apply the principles and methodologies of digital forensics to conduct effective investigations into cybercrimes and incidents.
- Students will be able to demonstrate proficiency in acquiring and analyzing digital evidence from a variety of devices and data sources.
- Students will be able to utilize forensic tools and techniques to recover, preserve, and present digital evidence in a legally sound manner.
- Students will be able to understand the legal and ethical considerations in digital forensics, including chain of custody and expert witness responsibilities.
- Students will be able to conduct practical digital forensic investigations, analyze case studies, and communicate findings effectively.

Green Computing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Green Computing
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Computer Organization and Operating Systems

Course Objectives

The course aims to:

- Introduce the fundamental concepts and principles of green and sustainable computing.
- Develop an understanding of energy-efficient computing systems, hardware, software, and data centres.
- Explain the role of virtualization, cloud computing, and resource management in reducing energy consumption and environmental impact.
- Introduce sustainable approaches to software development, electronic waste management, and the life cycle of computing systems.
- Familiarize students with carbon-aware computing, green metrics, and emerging approaches to sustainable digital infrastructure.

Syllabus

UNIT I

Fundamentals of Green Computing

Introduction to green computing; need for sustainable computing; environmental impact of Information and Communication Technology; energy consumption in computing systems; carbon footprint; energy efficiency and sustainability; green computing principles; life cycle of computing systems; sustainable hardware and software; energy-aware computing; green IT and sustainable IT; green computing standards and practices.

UNIT II

Energy-Efficient Computing Systems

Energy consumption in processors, memory, storage, and communication systems; power management techniques; dynamic voltage and frequency scaling; processor and memory optimization; energy-efficient operating systems; energy-aware scheduling; low-power computing; energy-efficient storage; energy-efficient networking; performance versus energy trade-offs; metrics for measuring energy efficiency.

UNIT III

Green Data Centres and Cloud Computing

Energy consumption in data centres; data centre infrastructure; server consolidation; virtualization and energy efficiency; workload management; resource allocation; energy-aware scheduling; cooling and thermal management; power usage effectiveness (PUE); energy-efficient storage and networking; green cloud computing; dynamic resource provisioning; cloud workload optimization; renewable energy-powered data centres.

UNIT IV

Green Software and Sustainable Computing

Green software principles; energy-aware software design; software efficiency and resource utilization; energy-aware algorithms; computational complexity and energy consumption; sustainable programming practices; application performance and energy trade-offs; green mobile and edge computing; energy-efficient Internet of Things systems; carbon-aware software development; measuring and optimizing the energy consumption of software.

UNIT V

E-Waste, Carbon-Aware Computing and Applications

Electronic waste and its environmental impact; sources and classification of e-waste; e-waste management and recycling; extended producer responsibility; sustainable procurement and hardware life extension; circular economy in computing; carbon-aware computing; carbon accounting and carbon intensity; renewable-energy-aware workload scheduling; sustainable artificial intelligence and computing; green computing applications; challenges and future directions in sustainable computing.

Text Books

Text Books

1. Bud E. Smith, *Green Computing: Tools and Techniques for Saving Energy, Money, and Resources*, CRC Press.
2. Jason H. H. L. and other contributors, *Green Computing: Tools and Techniques for Saving Energy, Money, and Resources*, CRC Press.

Reference Books

Reference Books

1. Jason W. B. and other contributors, *Green Computing and Sustainability*, Springer.
2. Bhuvan Unhelkar, *Green IT Strategies and Applications*, CRC Press.
3. Wu-chun Feng, *The Green Computing Book: Tackling Energy Efficiency at Large Scale*, CRC Press.

Course Outcomes

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

- Explain the fundamental concepts, principles, and environmental implications of green computing.
- Analyse energy consumption in computing systems and apply basic techniques for improving energy efficiency.
- Explain the role of virtualization, cloud computing, and resource management in developing energy-efficient data centres.
- Apply basic principles of energy-aware and sustainable software development.
- Analyse e-waste, carbon emissions, and sustainability challenges associated with modern computing systems and identify appropriate mitigation approaches.

Image Processing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Image Processing
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	NIL

Course Objectives

- To become familiar with digital image fundamentals.
- Be exposed to simple image processing techniques.
- To get exposed to simple image enhancement techniques in Spatial and Frequency domain.
- To learn concepts of degradation function and restoration techniques.
- To study the image segmentation and representation techniques.
- Learn to represent image in form of features.

Syllabus

UNIT I

Digital Image Fundamentals

Steps in Digital Image Processing, Components, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Relationships between pixels, neighborhood, adjacency, connectivity, distance measures, Color image fundamentals, RGB, HSI models.

UNIT II

Image Enhancements

Spatial Domain: Gray level transformations, Histogram processing, Basics of Spatial Filtering, Smoothing and Sharpening Spatial Filtering.

Frequency Domain: Introduction to Fourier Transform, Smoothing and Sharpening frequency domain filters, Ideal, Butterworth and Gaussian filters, Homomorphic filtering, Color image enhancement.

UNIT III

Image Restoration

Image Restoration, degradation model, Properties, Noise models, Mean Filters, Order Statistics, Adaptive filters, Band reject Filters, Band pass Filters, Notch Filters, Optimum Notch Filtering, Inverse Filtering, Wiener filtering.

UNIT IV

Image Segmentation and Color Image Processing

Edge detection, Edge linking via Hough transform, Thresholding, Region based segmentation, Region growing, Region splitting and merging, Morphological processing, erosion and dilation, Segmentation by morphological watersheds, basic concepts, Dam construction, Watershed segmentation algorithm. Color models, RGB, YUV, HSI; Color transformations, formulation, color complements, color slicing, tone and color corrections; Color image smoothing and sharpening; Color Segmentation.

UNIT V

Wavelets and Morphological Image Processing

Uncertainty principles of Fourier Transform, Time frequency localization, continuous wavelet transforms, wavelet bases and multi-resolution analysis, wavelets and Sub band filter banks, wavelet packets. Erosion and Dilation, Opening and Closing, The Hit-or-Miss Operation, Some Basic Morphological Algorithms, Boundary Extraction, Extraction of Connected Components, Convex Hull, Thinning and Thickening.

Text Books

Text Books

1. Rafael C. Gonzalez, Richard E. Woods,Digital Image Processing Pearson, Third Edition, 2010.
2. Anil K. Jain,Fundamentals of Digital Image Processing Pearson, 2002.

Reference Books

Reference Books

1. Kenneth R. Castleman,Digital Image Processing Pearson, 2006.
2. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins,Digital Image Processing using MATLAB Pearson Education, Inc., 2011.
3. D. E. Dudgeon and RM. Mersereau,Multidimensional Digital Signal Processing Prentice Hall Professional Technical Reference, 1990.
4. William K. Pratt,Digital Image Processing John Wiley, New York, 2002.
5. Milan Sonka et al Image processing, analysis and machine vision Brookes/Cole, Vikas Publishing House, 2nd edition, 1999.

Course Outcomes

- At the end of the course, the students should be able to:
- Know and understand the basics and fundamentals of digital image processing, such as digitization, sampling, quantization, and 2D-transforms.
- Operate on images using the techniques of smoothing, sharpening and enhancement.
- Understand the restoration concepts and filtering techniques.

- Learn the basics of segmentation, features extraction, compression and recognition methods for color models.

Introduction to Large Language Models

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Introduction to Large Language Models
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Programming Fundamentals, Data Structures and Basic Artificial Intelligence

Course Objectives

The course aims to:

- Introduce the fundamental concepts, architectures, and evolution of Large Language Models (LLMs).
- Develop an understanding of the basic components involved in training and using modern language models.
- Explain tokenization, embeddings, attention mechanisms, and the Transformer architecture underlying contemporary LLMs.
- Introduce fundamental techniques for adapting and interacting with LLMs, including prompting, fine-tuning, and retrieval-augmented generation.
- Familiarize students with the applications, limitations, evaluation, safety, and responsible use of Large Language Models.

Syllabus

UNIT I

Foundations of Large Language Models

Introduction to Natural Language Processing and language models; evolution of language models; statistical language models; neural language models; word representations and embeddings; recurrent neural networks and their limitations; emergence of Transformer-based language models; Large Language Models and foundation models; capabilities and limitations of LLMs; overview of major LLM families and applications.

UNIT II

Transformer Architecture

Sequence modelling and attention; self-attention and multi-head attention; queries, keys and values; positional encoding; Transformer encoder and decoder; feed-forward networks; residual connections and normalization; causal language modelling; autoregressive text generation; encoder-only, decoder-only, and encoder–decoder architectures; overview of Transformer- based language models.

UNIT III

Training and Adaptation of LLMs

Language model pre-training; next-token prediction; training data and data preparation; tokenization and subword representations; model parameters and context windows; computational requirements of LLM training; inference and text generation; temperature and sampling strategies; fine-tuning; instruction tuning; parameter-efficient adaptation; introduction to preference-based alignment; model adaptation for specific tasks and domains.

UNIT IV

Prompting, Retrieval and LLM Applications

Prompt engineering; zero-shot and few-shot prompting; role and instruction prompts; structured prompting; prompt templates; context management; retrieval-augmented generation (RAG); embeddings and vector databases; document retrieval and context construction; grounding language models using external knowledge; LLM applications in question answering, summarization, information extraction, code generation, and conversational systems; introduction to tool use and function calling.

UNIT V

Evaluation, Limitations and Responsible Use of LLMs

Evaluation of language models; accuracy and task-specific evaluation; perplexity and generation quality; factuality and hallucination; bias and fairness; toxicity and harmful content; prompt injection and basic security risks; privacy and data protection; copyright and intellectual property; interpretability and transparency; reliability and robustness; responsible use of LLMs; human oversight; limitations of current Large Language Models; emerging applications and future directions.

Text Books

Text Books

1. Sebastian Raschka, *Build a Large Language Model (From Scratch)*, Manning Publications.
2. Chakraborty, T. (2024). Introduction to Large Language Models: Generative AI for Text. Wiley India.

Reference Books

Reference Books

1. Daniel Jurafsky and James H. Martin, *Speech and Language Processing*, Pearson.
2. Chip Huyen, *AI Engineering: Building Applications with Foundation Models*, O'Reilly Media.
3. Jay Alammar and Maarten Grootendorst, *Hands-On Large Language Models: Language Understanding and Generation*, O'Reilly Media.

Course Outcomes

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

- Explain the evolution, fundamental concepts, architectures, and applications of Large Language Models.
- Describe the working principles of tokenization, embeddings, attention, and Transformer-based language models.
- Explain the basic processes involved in pre-training, inference, fine-tuning, and adaptation of Large Language Models.
- Develop basic LLM-based applications using prompting, retrieval- augmented generation, and external knowledge sources.
- Evaluate the capabilities and limitations of Large Language Models and identify issues related to hallucination, bias, security, privacy, and responsible use.

Mobile Computing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Mobile Computing
Category of Course	Professional Elective Courses/Open Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Computer Networks

Course Objectives

- To study the specifications and functionalities of various protocols/standards of mobile networks.
- To learn about the concepts and principles of mobile computing.
- To explore both theoretical and practical issues of mobile computing.
- To develop skills of finding solutions and building software for mobile computing applications.

Syllabus

UNIT I

Introduction

Challenges in mobile computing, Description of cellular system, Frequency Reuse, Co-channel and Adjacent channel interference, Propagation Models for Wireless Networks, Multipath Effects in Mobile Communication, channel allocation, Handoff, types of handoffs; location management.

UNIT II

Evolution of Modern Mobile Wireless Communication System

First Generation Wireless Networks, Second Generation (2G) Wireless Cellular Networks, Major 2G standards, GSM: Architecture and Protocols

,2.5G Wireless Networks, The General Packet Radio Services: (GPRS), Overview of CDMA systems: IS-95 Networks.

UNIT III

3G Mobile Networks, Cellular

WLAN Integration, Introduction to 4G, WiMAX, LTE, Mobile IP, Mobile TCP.

UNIT IV

Support for mobility

File systems, World Wide web, Wireless application protocol, Mobile operating systems, Mobile agents, Satellite Systems, Global Positioning System.

UNIT V

Layered architecture of MANET, Ad hoc network routing protocols, MAC and Transport layer issues of MANET, Introduction to Wireless Sensor Network ,Wireless Mesh Network , VANET

Text Books

Text Books

1. P.K. Pattnaik, Rajib Mall, “Fundamentals Of Mobile Computing”, PHI, 2015.
2. D.P. Agrawal and Q.A. Zeng, “Introduction to Wireless and Mobile Systems”,3rd edition, Thomson Learning, 2010.
3. J. Schiller, “Mobile Communications”, 2nd edition, Pearson Education, 2012.

Course Outcomes

- Have a good understanding of how the underlying wireless and mobile communication networks work, their technical features, and what kinds of applications they can support.
- Identify the important issues of developing mobile computing systems and applications.
- Develop mobile computing applications by analyzing their characteristics and requirements, selecting the appropriate computing models and software architectures, and applying standard programming languages and tools.
- Organize and manage software built for deployment and demonstration.

Natural Language Processing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Speech and natural Language Processing
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic Programming Skills General Understanding Of Statistics

Course Objectives

- This course introduces the fundamental concepts and techniques of natural language processing (NLP). It provides the understanding of the computational properties of natural languages and the algorithms for processing linguistic information. The course will introduce both linguistic (knowledge-based) and statistical approaches to NLP, illustrate the use of NLP techniques and tools in a variety of application areas, and provide insight into many open research problems.

Syllabus

UNIT I

Introduction

Introduction to NLP, challenges of NLP, Phases in natural language processing, applications Language Modeling: Grammar-based LM, Statistical LM Regular Expression, Finite State Automata,

Morphology and Finite State Transducers, N-grams, Smoothing, HMM and Speech Recognition: Speech Recognition Architecture, Overview of HMM. Evaluation of language models.

UNIT II

Word Classes and Part-of-Speech Tagging

English word classes, Targets for English, Part of speech Tagging, Rule Based part of speech Tagging, Stochastic part of speech Tagging,HMM,Transformation Based Tagging. Handling of unknown words, named entities, multi word expressions.

Context Free Grammars for English

Constituency, Context Free rules and Trees, Sentence level construction, The Noun Phrase, Coordination, Agreement, The verb phrase and sub-categorization. Parsing with context free grammars: Basic Top down Parser, and Bottom-up parsing, the early Algorithm, Finite state parsing method.

Features and Unifications

Feature structures, Unification of Features Structures, Features Structures in the grammar, Implementing Unification.

Lexicalized and probabilistic parsing

Probabilistic context free grammars, problems with probabilistic context free grammars, probabilistic lexicalized GFG.

UNIT III

Semantics Representing Meaning

Meaning structure of language, First order predicate calculus, linguistically relevant concept, Related Re-presentational approaches, Alternative approaches to meaning.

Semantic Analysis

Syntax driven semantic analysis, Attachment of Fragment of English,. Robust Semantic Analysis.

Lexical Semantics

Relation among lexemes and their senses, Internal Structure of words. WordNet, Word Sense Disambiguation- Selectional restriction, machine learning approaches, dictionary based approaches.

UNIT IV

Pragmatics Discourse

Reference resolution, Text Coherence, Discourse Structure, constraints on co-reference algorithm for pronoun resolution Psycholinguistics Studies of reference and coherence. Natural Language generation: Introduction to language generation, Architecture for generation,, Discourse planning.

UNIT V

Applications of NLP

Introduction to corpus elements in balanced corpus, TreeBank, PropBank, WordNet, VerbNet etc. stemmers and lemmatiser, Spell-checking, Summarization Information Retrieval- Vector space model, term weighting, Machine Translation– Overview.

Reference Books

Reference Books

1. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008. Tanveer Siddiqui, U.S. Tiwary, —Natural Language Processing and Information Retrieval, Oxford University Press, 2008.
2. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming, 1995.
3. Manning, Christopher and Heinrich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
4. Richard M Reese, —Natural Language Processing with Java, OReilly Media, 2015.
5. Nitin Indurkhya and Fred J. Damerau, —Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC Press, 2010.

Course Outcomes

- Understand core algorithms and data structures used in NLP.
- Apply these mathematical models and algorithms in applications in software design and implementation for NLP.
- Develop NLP components, such as n-gram language models stemmer, part-of-speech taggers.
- Evaluate the merits of use of different statistical approaches for different types NLP tasks.
- Implement a simple NLP systems.

Network on Chip

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Network on Chip
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Computer Networks and Data Communication

Course Objectives

- To introduce basic interconnection networks and its various uses.
- To introduce popular topologies in NoC.
- To introduce Routing and routing mechanics in NoC.
- To introduce flow control and QoS primitives in NoC.

Syllabus

UNIT I

Introduction to Interconnection Networks

Uses of Interconnection Networks, Network Basics:Topology, Routing Flow Control, Router Architecture, Performance of Interconnection Networks.

UNIT II

Topology Basics

Channels and Nodes, Direct and Indirect Networks, Cuts and Bisections, Paths, Traffic Patterns, Performance, Throughput and Maximum Channel Load, Latency, Path Diversity, Packaging Cost.

Basics of Popular Topologies:

Butterfly Networks, Structure, Performance, Packaging cost, Path diversity, Number of Stages; Torus Networks, Structure, Performance, Packaging cost, Path diversity.

Meshes and Express cubes.

UNIT III

Routing Basics

Taxonomy of Routing Algorithms, The Routing Relation, Deterministic Routing, Oblivious Routing, Minimal Oblivious Routing, Load Balanced Oblivious Routing, Adaptive Routing, Routing Mechanics.

UNIT IV

Flow Control Basics Resources and Allocation Units, Buffer less Flow Control, Circuit Switching, Buffered Flow Control, Packet-Buffer Flow Control,Flit-Buffer,Flow Control, Buffer Management and Backpressure, Flit-Reservation Flow Control.

UNIT V

Deadlock and Livelock

Deadlock, Deadlock Avoidance, Adaptive Routing, Deadlock Recovery, Livelock;Quality of Service, Burstiness and Network, Implementation of Guaranteed Services, Delays, Implementation of Best-Effort Services, Separation of Resources.

Text Books

Text Books

1. “Principle and Practices of Interconnection Networks”, William J. Dally and brain Towles.Morgankaufmann.

Reference Books

Reference Books

1. “Network – On – Chip: From Implementation to programming Paradigm”, Sheng Ma Libo Huang Mingche Lai Wei Shi, Margan Kaufman.

Course Outcomes

- In depth analysis of Commercial NoCs.
- Understanding of basic requirements of NoC topologies and various performance factors.
- Understanding how to avoid deadlocks and lovelocks in various choices of routing algorithms present.
- Understanding the QoS requirements.

Distributed Systems

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Distributed Systems
Category of Course	Professional Elective Courses
Credits	3L:0T: 0P C:3
Pre-Requisites	Operating Systems

Course Objectives

- Introduce students to the general properties, characteristics, and issues of distributed systems. Also, students should be able to understand how modern systems works.
- Students should be able to learn on distributed algorithms and how these algorithms are applied when designing and implementing real systems.
- Students learned about some topics on clock synchronization, coordination algorithms, transactions, and replications.
- Students should be able to understand design and implementation issues on distributed shared memory.
- Students should be able to learn experimental experience in designing and implementing real systems through computer-based assignments.

Syllabus

UNIT I

Introduction

Background: Brief definitions of distributed systems, Motivation, Examples of distributed Systems, Relation to parallel systems, Message passing systems Vs. Shared memory systems, Execution process for synchronous Vs. asynchronous, Case Study, World Wide Web.

UNIT II

Communications in Distributed Computing Models

System Models: Architectural models, Interaction model, Failure model, security model. Inter process communication: API for the internet protocols, External data representation and Marshaling, Client-Server communication, and Group communication, Message queues, Case study: Interprocess communication in Unix. Distributed objects and Remote invocation: Distributed objects, Communication between distributed objects, Request-Reply protocols, Remote procedure call, Remote method invocation, Case study: Java RMI.

UNIT III

Peer-to-Peer services and File systems Peer-to-Peer systems: Introduction, Napster and its legacy, Peer-to-peer middleware, Routing overlays, Case study: Pastry, Tapestry. Distributed File systems: Introduction, File service architecture, Andrew file system. Name services: Introduction, Name services

and the Domain Name System, directory services, Case study: The Global Name Service.

UNIT IV

Synchronization and Replication

Time and Global States: Introduction, Clocks, events, and process states, Synchronizing physical clocks, Logical time and logical clocks, Global states. Coordination and Agreement: Introduction, Distributed mutual exclusion, Elections, Coordination and agreement in group communication. Transaction and Concurrency control: Transactions, Nested transactions, Locks, Optimistic concurrency control, Timestamp ordering.

Distributed Transactions: Introduction, Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery, Replication.

UNIT V

Process and Resource Management

Process Management: Process migration, Features, Mechanism, and Threads: Models, Issues, Implementation. Resource Management: Introduction, Scheduling Algorithms, Task Assignment Approach, Load Balancing Approach, Load Sharing Approach.

Text Books

Text Books

1. Distributed Systems Concept and Design, 5th Edition, George Coulouris, Jean Dollimore, Tim Kindberg, published by Pearson Education, Copyright © 2012.

Reference Books

Reference Books

1. Distributed Operating Systems: Concepts and Design, Pradeep K Sinha, published by Prentice Hall of India, 2007.
2. Distributed computing: principles, algorithms, and systems, Kshemkalyani, Ajay D., and Mukesh Singhal, published by Cambridge University Press, 2011.
3. Distributed Systems: Principles and Paradigms, 3rd Edition, Tanenbaum A.S., Van Steen M., published by Pearson Education, 2017.

Course Outcomes

- Students will be able to apply the concept of distributed systems, techniques, and trends.
- Students will be capable of applying the concept of network virtualization, remote method invocation, and distributed objects.

- Students will be able to understand the peer-to-peer services, distributed file systems, domain name system.
- Students will gain the knowledge of logical clocks, distributed mutual exclusion, distributed deadlocks, concurrency control in distributed transactions, replications.
- Students will be able to capture the knowledge of process and resource management.

Quantum Computing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Quantum Computing
Category of Course	Professional Elective Courses
Credits	3L:0T: 0P C:3
Pre-Requisites	Linear Algebra Basics

Course Objectives

- The course will provide an insight of basic of quantum physics from a computer scientist's perspective, and how it describes reality and understand the philosophical implications of quantum computing.

Syllabus

UNIT I

Qubit& Quantum States

The Qubit, Vector Spaces. Linear Combination of Vectors, Uniqueness of a spanning set, basis & dimensions, inner Products, orthonormality, Gram-Schmidt orthogonalization, bra-ketformalism, the Cauchyschwarez and triangle Inequalities.

UNIT II

Matrices & Operators

Observables, The Pauli Operators, Outer Products, The Closure Relation, Representation of operators using matrices, outer products & matrix representation, matrix representation of operators in two dimensional spaces, Pauli Matrix, Hermitian unitary and normal operator, Eigen values & Eigen Vectors, Spectral Decomposition, Trace of an operator, important properties of Trace, Expectation Value of Operator, Projection Operator, Positive Operators.

UNIT III

Commutator Algebra

Heisenberg uncertainty principle, polar decomposition &singular values, Postulates of Quantum Mechanics. Tensor Products: Representing Composite States in Quantum Mechanics, Computing inner products, Tensor products of column vectors, operators and tensor products of Matrices.

UNIT IV

Density Operator Density Operator of Pure & Mix state, Key Properties, Characterizing Mixed State, Practical Trace & Reduce Density Operator, Density Operator & Bloch Vector. Quantum Measurement Theory: Distinguishing Quantum states & Measures, Projective Measurements,

Measurement on Composite systems, Generalized Measurements, Positive Operator- Valued Measures

UNIT V

Recent trends in Quantum Computing Research

Quantum Computing Applications of soft computing. Quantum Cryptography, Quantum Automata Theory etc.

Text Books

Text Books

1. M. A. Nielsen and I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2000.
2. Quantum Computing Explained By DAVID Mc MAHON

Reference Books

Reference Books

1. Quantum Computing without Magic by ZdzislawMeglicki
2. Quantum Computer Science By Marco Lanzagorta, Jeffrey Uhlmann
3. . An Introduction to Quantum Computing Phillip Kaye, Raymond Laflamme, Michele Mosca.

Course Outcomes

- Knowledge of Vector spaces, Matrices, Quantum state, Density operator and Quantum Measurement theory.
- Application of quantum computing to soft computing and Cryptography.

Recommendation Systems

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Recommendation Systems
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Programming Fundamentals, Data Structures and Basic Machine Learning

Course Objectives

The course aims to:

- Introduce the fundamental concepts, architecture, and applications of recommendation systems.
- Develop an understanding of user preferences, item representation, ratings, and user–item interactions.
- Introduce content-based and collaborative filtering techniques for generating recommendations.
- Familiarize students with basic model-based, hybrid, and context-aware recommendation approaches.
- Develop an understanding of recommendation system evaluation, challenges, and responsible recommendation practices.

Syllabus

UNIT I

Introduction to Recommendation Systems

Introduction and motivation; information filtering and personalization; recommendation problem; users, items, preferences and interactions; explicit and implicit feedback; rating data; user–item matrix; recommendation system architecture; non-personalized and personalized recommendations; popularity- based recommendation; basic data preprocessing; applications of recommendation systems in e-commerce, entertainment, social media, education and information services.

UNIT II

Content-Based Recommendation

Fundamentals of content-based filtering; item representation and item profiles; user profiles; feature extraction; categorical and textual features; vector representation; TF–IDF; similarity measures; cosine similarity; profile construction and recommendation generation; advantages and limitations of content-based recommendation; basic implementation of a content-based recommender.

UNIT III

Collaborative Filtering

Fundamentals of collaborative filtering; user-based and item-based collaborative filtering; neighbourhood-based methods; similarity between users and items; Pearson correlation and cosine similarity; nearest neighbours; rating prediction; top-N recommendation; sparsity problem; cold-start problem; introduction to model-based collaborative filtering; matrix factorization and latent factors.

UNIT IV

Hybrid and Context-Aware Recommendation

Limitations of individual recommendation approaches; hybrid recommendation systems; combining content-based and collaborative filtering; weighted, switching and mixed hybrid approaches; introduction to context-aware recommendation; contextual information such as time, location and user situation; introduction to knowledge-based recommendation; basic concepts of session-based and sequential recommendation; overview of machine learning and neural approaches to recommendation.

UNIT V

Evaluation and Challenges of Recommendation Systems

Evaluation of recommendation systems; training and test data; offline and online evaluation; prediction accuracy; Mean Absolute Error and Root Mean Squared Error; Precision, Recall and F1-score; ranking metrics; Precision@K and Recall@K; coverage, diversity, novelty and serendipity; cold-start and data sparsity; scalability; popularity bias and filter bubbles; privacy and fairness; explainable recommendations; responsible recommendation systems; emerging applications and trends.

Text Books

Text Books

1. Charu C. Aggarwal, *Recommender Systems: The Textbook*, Springer.
2. Dietmar Jannach, Markus Zanker, Alexander Felfernig and Gerhard Friedrich, *Recommender Systems: An Introduction*, Cambridge University Press.

Reference Books

Reference Books

1. Francesco Ricci, Lior Rokach and Bracha Shapira, *Recommender Systems Handbook*, Springer.
2. Kim Falk, *Practical Recommender Systems*, Manning Publications.
3. Deepak K. Agarwal and Bee-Chung Chen, *Statistical Methods for Recommender Systems*, Cambridge University Press.

Course Outcomes

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

- Explain the fundamental concepts, components, and applications of recommendation systems.
- Develop basic content-based recommendation systems using item features, user profiles, and similarity measures.
- Apply user-based and item-based collaborative filtering techniques for rating prediction and recommendation.
- Explain basic model-based, hybrid, and context-aware approaches to recommendation.
- Evaluate recommendation systems using appropriate metrics and identify challenges related to sparsity, cold start, scalability, privacy, fairness, and bias.

Reinforcement Learning

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Reinforcement Learning
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Probability and Statistics, Data Structures and Basic Machine Learning

Course Objectives

The course aims to:

- Introduce the fundamental concepts and principles of reinforcement learning and sequential decision-making.
- Develop an understanding of Markov Decision Processes, rewards, policies, value functions, and the exploration–exploitation problem.
- Introduce fundamental value-based and policy-based reinforcement learning algorithms.
- Enable students to formulate basic decision-making problems as reinforcement learning problems and apply appropriate algorithms.
- Familiarize students with practical applications, challenges, and limitations of reinforcement learning.

Syllabus

UNIT I

Introduction to Reinforcement Learning

Introduction to reinforcement learning; machine learning paradigms; supervised, unsupervised and reinforcement learning; characteristics of reinforcement learning; agent, environment, state, action and reward; interaction between agent and environment; episodes and trajectories; return and cumulative reward; exploration and exploitation; examples of reinforcement learning problems; applications in games, robotics, recommendation, resource management and autonomous systems.

UNIT II

Markov Decision Processes

Markov property; Markov chains and Markov Decision Processes; states, actions, rewards and transition probabilities; policies; stochastic and deterministic policies; reward functions; discount factor; value functions; state-value and action-value functions; Bellman equations; optimal value functions and optimal policies; formulation of practical problems as Markov Decision Processes.

UNIT III

Value-Based Reinforcement Learning

Dynamic programming for reinforcement learning; policy evaluation; policy improvement; policy iteration; value iteration; Monte Carlo methods; temporal-difference learning; TD prediction; SARSA; Q-learning; on-policy and off-policy learning; exploration strategies; ϵ-greedy method; advantages and limitations of value-based methods.

UNIT IV

Function Approximation and Policy-Based Learning

Need for function approximation; tabular versus approximate reinforcement learning; linear function approximation; introduction to neural networks for reinforcement learning; Deep Q-Networks (DQN); experience replay and target networks; policy-based reinforcement learning; policy representation; policy gradients; actor–critic approach; basic comparison of value-based and policy-based methods.

UNIT V

Applications, Evaluation and Challenges

Reinforcement learning applications in games, robotics, recommendation systems, healthcare, resource allocation and autonomous systems; environment design and reward engineering; training and evaluation of RL agents; sample efficiency; convergence and stability; reward sparsity; exploration challenges; safety and reliability; ethical considerations; introduction to advanced reinforcement learning methods; current trends and future directions.

Text Books

Text Books

1. Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd Edition, MIT Press.
2. Csaba Szepesvári, *Algorithms for Reinforcement Learning*, Springer.

Reference Books

Reference Books

1. Marco Wiering and Martijn van Otterlo, *Reinforcement Learning: State-of-the-Art*, Springer.
2. Laura Graesser and Wah Loon Keng, *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*, Addison-Wesley.
3. Maxim Lapan, *Deep Reinforcement Learning Hands-On*, 2nd Edition, Packt Publishing.

Course Outcomes

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

- Explain the fundamental concepts of reinforcement learning, sequential decision-making, and agent–environment interaction.
- Formulate decision-making problems using Markov Decision Processes, reward functions, policies, and value functions.
- Apply fundamental reinforcement learning algorithms including dynamic programming, Monte Carlo methods, SARSA, and Q-learning.
- Explain the basic principles of deep and policy-based reinforcement learning and distinguish between major reinforcement learning approaches.
- Analyse practical challenges and applications of reinforcement learning, including exploration, reward design, sample efficiency, safety, and reliability.

Robotics

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Robotics
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Engineering Mathematics, Programming Fundamentals and Basic Mechanics

Course Objectives

The course aims to:

- Introduce the fundamental concepts, components, and classifications of robotic systems.
- Develop an understanding of robot kinematics, coordinate transformations, and motion representation.
- Introduce the basic principles of robot dynamics, actuators, sensors, and motion control.
- Familiarize students with robot perception, localization, path planning, and navigation.
- Introduce the applications of robotics and the role of artificial intelligence in modern robotic systems.

Syllabus

UNIT I

Introduction to Robotics

Introduction to robotics; history and evolution of robots; characteristics and applications of robots; robot classification; industrial and service robots; robot components; links, joints and manipulators; degrees of freedom; robot workspace; end-effectors; actuators; sensors; robotic control systems; serial and parallel robots; mobile robots; overview of modern autonomous robots.

UNIT II

Robot Kinematics

Coordinate systems and transformations; position and orientation of rigid bodies; rotation matrices; homogeneous transformations; robot configuration; forward kinematics; Denavit–Hartenberg representation; inverse kinematics; multiple solutions and singularities; velocity kinematics; Jacobian matrix; basic motion representation and trajectory concepts.

UNIT III

Robot Dynamics and Control

Introduction to robot dynamics; forces and torques; equations of motion; robot actuators; DC motors, servo motors and stepper motors; basic robot control concepts; position and velocity control; feedback control; proportional–integral–derivative (PID) control; trajectory tracking; joint-space and Cartesian-space control; stability and basic performance considerations.

UNIT IV

Robot Perception, Localization and Planning

Role of sensors in robotics; proprioceptive and exteroceptive sensors; cameras, LiDAR, ultrasonic and inertial sensors; basic computer vision for robotics; robot localization; mapping; introduction to Simultaneous Localization and Mapping (SLAM); configuration space; obstacle avoidance; path planning; graph-based search; A* algorithm; basic motion planning; navigation of mobile robots.

UNIT V

Intelligent and Autonomous Robotics

Autonomous robotic systems; robot operating environments; introduction to Robot Operating System (ROS); robot simulation; human–robot interaction; robot learning and artificial intelligence; basic concepts of reinforcement learning in robotics; collaborative robots; autonomous mobile robots; robotics in manufacturing, healthcare, agriculture, logistics and exploration; safety and ethical considerations; emerging trends in intelligent robotics.

Text Books

Text Books

1. John J. Craig, *Introduction to Robotics: Mechanics and Control*, 4th Edition, Pearson.
2. Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani and Giuseppe Oriolo, *Robotics: Modelling, Planning and Control*, Springer.

Reference Books

Reference Books

1. Kevin M. Lynch and Frank C. Park, *Modern Robotics: Mechanics, Planning, and Control*, Cambridge University Press.
2. Roland Siegwart, Illah R. Nourbakhsh and Davide Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd Edition, MIT Press.
3. Peter Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, Springer.

Course Outcomes

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

- Explain the fundamental components, classifications, architectures, and applications of robotic systems.
- Apply coordinate transformations and kinematic principles to determine the position and orientation of robotic manipulators.
- Explain the fundamental principles of robot dynamics, actuators, feedback control, and trajectory tracking.
- Apply basic techniques for robot perception, localization, path planning, and navigation.
- Explain the role of artificial intelligence in autonomous robotics and identify major applications, challenges, safety issues, and emerging trends in robotics.

Social Network Analysis

Course Code	As per the choice of semester and paper
Course Name	Social Network Analysis
Credits	3L:0T: 0 P (3 Credits)
Pre-Requisites	Wireless Networks(Desirable)

Course Objectives

- To Understand the fundamental concepts and theories of social network analysis. To understand the concepts of link prediction, community detection etc for social network analysis To Learn the methods and techniques used for analyzing and visualizing social networks. To Apply social network analysis to real-world data sets.

Syllabus

UNIT I

Hours

INTRODUCTION:

Overview of Networks: What is a social network? Types of networks, Nodes, edges, degree, Types of network structures: Random, small-world, scale-free

History and Evolution of SNA, Introduction to social network mining. Illustration of various social network mining tasks with real-world examples. Introduction to Semantic Web, Relationship between Social Networks, Semantic Web and Knowledge Graphs, Evolution from Social Networks to Graph Data Science

Applications of SNA: Social media analysis, recommendation systems, and organizational networks.

UNIT II

Graph Theory Fundamentals: Graphs, directed and undirected graphs, weighted and unweighted graphs, Types of graphs: Bipartite graphs, trees, and multi-graphs. Network Representation: Adjacency matrix, adjacency list, edge list, Visualizing networks.

Network Models: Random graphs (Erdős–Rényi model).,Small-world networks (Watts-Strogatz model).,Scale-free networks (Barabási–Albert model).

Network Topology: Clustering, connectedness, components, diameter, and radius.

UNIT III

Centrality Measures: Degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality.

Node and Edge Metrics: Local vs. global metrics. Clustering coefficient, assortativity, and transitivity.

Graph and Network Metrics, Path length, diameter, and distance distribution,Network density and modularity.

Applications of Centrality: Identifying influencers in social media, Central nodes in communication networks.

Link Prediction in Online Social Network, Cascade Behaviors and Network Effects

UNIT IV

Community Detection in Social Networks, Applications of Community detection, types of Community, community detection Algorithms: Modularity-based approaches (Louvain method).Spectral clustering, hierarchical clustering.

Web search algorithms,: PageRank, personalized PageRank, HITS Algorithm,

Case Studies: Identifying communities in social media platforms (e.g., Facebook, Twitter).

UNIT V

Real-World Applications: Social media analysis (Twitter, Facebook etc).

Tools for SNA: Introduction to network analysis tools: Gephi, NetworkX (Python), Cytoscape.

Trust and Security Issues in social networks

Case Studies and Practical Applications: Social network analysis for marketing, politics, and e-commerce,Crisis management, fraud detection, and intelligence networks.

Reference Books

Reference Books

1. Tanmoy Chakraborty, Social Network Analysis, Wiley,2021 Stanley Wasserman Social Network Analysis Methods and Applications, Cambridge University Pres,2012 A. L. Barabási, Network Science, Cambridge University Press, 2016 John Scott Social Network Analysis: A Handbook, SAGE Publications, 2017 David Easley & Jon Kleinberg – Networks, Crowds and Markets Mark Newman – Networks

Course Outcomes

- understand the basic concepts of social networks understand the concepts of link prediction and its applications understand the concepts of community detection and its applications understand the fundamental concepts in analyzing the large-scale data that are derived from social networks understand the security issues in social networks and probable measures to prevent attackes

Soft Computing

Course Code	Code to be assigned as per semester/paper offered (Professional Elective / Open Elective)
Course Name	Soft Computing
Category of Course	Professional Elective Course (PE) / Open Elective (OE)
Credits	3L : 0T : 0P C : 3
Pre-Requisites	Machine Learning (Desirable)

Course Objectives

- To introduce the concept of soft computing and distinguish it from conventional (hard) computing. To study artificial neural networks and their learning paradigms. To introduce fuzzy sets, fuzzy logic, and fuzzy inference systems for reasoning under uncertainty. To introduce evolutionary and swarm-based optimization techniques – genetic algorithms and particle swarm optimization. To develop practical skills in implementing soft computing techniques using Python-based tools. To expose students to current research trends and hybrid soft computing approaches.

Syllabus

UNIT I

- Introduction to Soft Computing and Neural Networks

Hours

Evolution of computing – hard computing vs soft computing, soft computing constituents; from conventional AI to computational intelligence.

Machine learning basics recap; learning using neural networks – adaptive networks, feedforward networks, supervised learning neural networks, radial basis function networks.

Overview of reinforcement learning and unsupervised learning neural networks; adaptive resonance architectures; recent advances in neural networks.

UNIT II

- Fuzzy Logic

Fuzzy sets and operations on fuzzy sets, fuzzy relations, membership functions.

Fuzzy rules and fuzzy reasoning, fuzzy inference systems (Mamdani and Sugeno models).

Fuzzy expert systems and fuzzy decision making; applications of fuzzy logic in control and pattern recognition.

UNIT III

- Evolutionary and Swarm-Based Optimization

Introduction to genetic algorithms (GA) – representation, selection, crossover, mutation; applications of GA in machine learning and knowledge acquisition.

Single-objective and multi/many-objective optimization; introduction to Pareto optimality.

Swarm intelligence – particle swarm optimization (PSO) and an overview of ant colony optimization (ACO); comparison with GA-based approaches.

UNIT IV

- Tools and Implementation

Python libraries for soft computing – scikit-fuzzy for fuzzy systems, DEAP/PyGAD for evolutionary algorithms, and NumPy/TensorFlow/Keras for neural network implementation.

Hands-on implementation of a simple artificial neural network, a fuzzy inference system, and a genetic algorithm for an optimization problem.

Overview of MATLAB Neural Network and Fuzzy Logic toolboxes as an alternative environment.

UNIT V

- Recent Trends and Hybrid Approaches

Neuro-fuzzy systems – adaptive neuro-fuzzy inference systems (ANFIS).

Hybrid soft computing approaches combining neural networks, fuzzy logic, and evolutionary algorithms.

Recent trends – soft computing in deep learning pipelines, explainable AI connections, and applications in optimization, control, and pattern classification.

Implementation and discussion of a recently proposed soft computing technique from current literature.

Text Books

Text Books

1. 1. J.-S.R. Jang, C.-T. Sun, E. Mizutani, “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”, Pearson Education.
2. 2. S.N. Sivanandam, S.N. Deepa, “Principles of Soft Computing”, 3rd Edition, Wiley India.

Reference Books

Reference Books

1. 1. S. Rajasekaran, G.A. Vijayalakshmi Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications”, PHI.
2. 2. George J. Klir, Ute St. Clair, Bo Yuan, “Fuzzy Set Theory: Foundations and Applications”, Prentice Hall.
3. 3. James A. Freeman, David M. Skapura, “Neural Networks: Algorithms, Applications and Programming Techniques”, Addison-Wesley.

4. 4. Kennedy & Eberhart, “Swarm Intelligence”, Morgan Kaufmann.

Course Outcomes

- On completion of this course, students will be able to: Distinguish soft computing from conventional computing and describe its major constituent techniques. Design and train artificial neural networks for classification and regression problems. Apply fuzzy logic and fuzzy inference systems to reason under uncertainty and solve engineering problems. Apply genetic algorithms and swarm-based optimization techniques to single-and multi-objective optimization problems. Implement soft computing techniques using Python-based tools to solve real-world problems. Analyze current research trends and hybrid approaches (e.g. neuro-fuzzy systems) in soft computing.

Software Testing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Software Testing
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Software Engineering and Object-Oriented Programming

Course Objectives

The course aims to:

- Introduce the fundamental concepts, principles, and processes of software testing.
- Develop an understanding of different testing levels, techniques, test design methods, and test case development.
- Introduce automated software testing and the role of testing tools in modern software development.
- Familiarize students with specialized testing approaches including integration, system, performance, security, and web application testing.
- Develop an understanding of software quality, test management, defect management, and modern continuous testing practices.

Syllabus

UNIT I

Fundamentals of Software Testing

Introduction to software testing; objectives and principles of testing; verification and validation; software errors, faults and failures; testing and debugging; software testing life cycle; test planning; test strategy; test levels; unit testing, integration testing, system testing and acceptance testing; functional and non-functional testing; regression testing; smoke and sanity testing; test cases, test suites and test scenarios; defect life cycle; role of testing in the software development life cycle.

UNIT II

Test Design Techniques

Black-box testing; equivalence partitioning; boundary value analysis; decision table testing; state transition testing; use-case testing; combinatorial testing; white-box testing; statement coverage; branch coverage; condition coverage; path testing; control-flow graphs; basic code-based test design; test data generation; exploratory testing; advantages and limitations of different test design techniques.

UNIT III

Test Automation and Automation Frameworks

Introduction to test automation; need for automation; automated versus manual testing; automation strategy and test selection; test automation frameworks; data-driven and keyword-driven testing; unit testing frameworks; automated functional testing; web application testing; API testing; test scripts and reusable components; test execution and reporting; continuous integration and automated testing; introduction to commonly used testing tools and frameworks; challenges and limitations of test automation.

UNIT IV

Specialized Software Testing

Integration and system testing; object-oriented software testing; web and mobile application testing; performance testing; load, stress and endurance testing; usability testing; compatibility testing; security testing; basic vulnerability-oriented testing; database testing; API and service testing; distributed application testing; testing of cloud-based software; introduction to testing Internet of Things applications.

UNIT V

Software Quality and Modern Testing Practices

Software quality assurance and software testing; quality metrics; defect tracking and defect management; test reporting; test effectiveness and coverage; risk-based testing; test management; configuration management; continuous testing in DevOps; shift-left and shift-right testing; testing in Agile development; exploratory and session-based testing; introduction to AI-assisted software testing; test generation and defect prediction; challenges in testing AI-based systems; emerging trends and best practices in software testing.

Text Books

Text Books

1. Paul Jorgensen, *The Craft of Software Testing: A Developer's Guide to Building Better Software*, CRC Press.
2. Srinivasan Desikan and Gopalaswamy Ramesh, *Software Testing: Principles and Practices*, Pearson.

Reference Books

Reference Books

1. Glenford J. Myers, Corey Sandler and Tom Badgett, *The Art of Software Testing*, 3rd Edition, Wiley.
2. Ron Patton, *Software Testing*, 2nd Edition, Sams Publishing.
3. Mark Fewster and Dorothy Graham, *Software Test Automation: Effective use of test execution tools*, Addison-Wesley.

Course Outcomes

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

- Explain the fundamental principles, processes, levels, and types of software testing.
- Design effective test cases using black-box and white-box testing techniques.
- Apply basic test automation techniques and use appropriate testing frameworks and tools.
- Explain and apply specialized testing approaches for web, performance, security, mobile, database, and distributed applications.
- Evaluate software quality using testing metrics and apply modern testing practices including risk-based, Agile, DevOps, and AI-assisted testing.

Speech Processing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Speech Processing
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Digital Signal Processing, Probability and Statistics, and Basic Machine Learning

Course Objectives

The course aims to:

- Introduce the fundamental characteristics of speech signals and the basic principles of speech processing.
- Develop an understanding of speech signal representation, sampling, analysis, and feature extraction.
- Introduce fundamental techniques for speech enhancement, segmentation, and speech analysis.
- Explain the basic principles and computational approaches used in automatic speech recognition.
- Familiarize students with major speech processing applications and recent developments based on machine learning and deep learning.

Syllabus

UNIT I

Fundamentals of Speech and Speech Signal Processing

Introduction to speech processing; human speech production and perception; speech production mechanism; acoustic characteristics of speech; voiced and unvoiced speech; speech signals and their properties; speech sampling and quantization; analog-to-digital conversion; time-domain and frequency-domain representation; Fourier transform; short-time analysis of speech; speech spectrograms; basic speech processing pipeline.

UNIT II

Speech Analysis and Feature Extraction

Short-time energy; zero-crossing rate; fundamental frequency and pitch; formants; autocorrelation analysis; linear predictive coding; cepstral analysis; Mel scale and Mel-frequency cepstrum; Mel-Frequency Cepstral Coefficients (MFCCs); filter banks; spectral features; prosodic features; feature normalization; speech feature representation for machine learning.

UNIT III

Speech Enhancement and Analysis

Speech enhancement and noise reduction; types and characteristics of speech noise; signal-to-noise ratio; filtering techniques; spectral subtraction; noise estimation; voice activity detection; speech segmentation; endpoint detection; echo and reverberation; basic speech separation; robust speech feature extraction; applications of speech enhancement.

UNIT IV

Automatic Speech Recognition

Introduction to Automatic Speech Recognition (ASR); speech recognition pipeline; acoustic, pronunciation and language models; template-based and statistical approaches; Hidden Markov Models; Gaussian Mixture Models; decoding and speech recognition; language modelling; word error rate; introduction to neural speech recognition; connectionist approaches; end-to-end speech recognition; basic concepts of Connectionist Temporal Classification and sequence-to-sequence models.

UNIT V

Speech Applications and Modern Speech Processing

Speech synthesis and Text-to-Speech systems; speaker recognition and speaker verification; speech emotion recognition; language and accent identification; keyword spotting; voice assistants and conversational systems; speech processing for multilingual and low-resource languages; introduction to self-supervised speech representation learning; neural speech processing; applications of speech processing in healthcare, education, accessibility and human–computer interaction; challenges and emerging trends in speech technology.

Text Books

Text Books

1. Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of Speech Recognition*, Prentice Hall.
2. Douglas O'Shaughnessy, *Speech Communications: Human and Machine*, 2nd Edition, Wiley.

Reference Books

Reference Books

1. Lawrence R. Rabiner and Ronald W. Schafer, *Theory and Applications of Digital Speech Processing*, Pearson.
2. Xuedong Huang, Alex Acero and Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall.
3. Ben Gold and Nelson Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, Wiley.

Course Outcomes

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

- Explain the fundamental characteristics, production mechanisms, and representations of speech signals.
- Apply basic signal processing and feature extraction techniques for speech analysis.
- Explain fundamental approaches for speech enhancement, noise reduction, segmentation, and robust speech processing.
- Describe the fundamental principles and components of automatic speech recognition systems.
- Explain major speech processing applications and evaluate the role of machine learning and deep learning in modern speech technologies.

User Interface Engineering

Course Code	Code will be given as per choice in particular semester and paper
Course Name	User Interface Engineering
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Programming Fundamentals and Object-Oriented Programming

Course Objectives

The course aims to:

- Introduce the fundamental principles of user interface engineering and modern web-based interface development.
- Develop proficiency in JavaScript programming for creating interactive and dynamic user interfaces.
- Explain the Document Object Model, event-driven programming, asynchronous JavaScript, and browser-based application development.
- Introduce component-based interface development using modern JavaScript frameworks and libraries.
- Develop an understanding of responsive, accessible, usable, and maintainable user interfaces and their testing.

Syllabus

UNIT I

Foundations of User Interface Engineering

Introduction to user interfaces and user interface engineering; principles of effective user interfaces; usability and user experience; user-centred design; interface structure and navigation; HTML5 fundamentals; semantic HTML; forms and input elements; CSS fundamentals; selectors, box model, layout and positioning; Flexbox and Grid; responsive design; media queries; web accessibility fundamentals; browser developer tools.

UNIT II

JavaScript Fundamentals for User Interfaces

JavaScript language fundamentals; variables, data types and operators; functions and scope; arrays and objects; destructuring; template literals; spread and rest operators; modules; classes and objects; error handling; higher-order functions; callbacks; functional programming concepts; Document Object Model (DOM); DOM traversal and manipulation; dynamic content generation; event handling; event propagation; form validation; browser events; local storage and session storage; basic client-side state management.

UNIT III

Asynchronous JavaScript and Web APIs

Synchronous and asynchronous programming; callbacks, Promises and async/await; asynchronous event handling; Fetch API; HTTP fundamentals; RESTful APIs; JSON; consuming web services from JavaScript; error handling for asynchronous operations; loading and error states; browser APIs; geolocation and storage APIs; timers; web workers; modules and package management; npm and JavaScript-based development tools; basic application bundling and development workflows.

UNIT IV

Component-Based User Interface Development

Introduction to component-based UI development; reusable interface components; component lifecycle; properties and state; event-driven components; conditional rendering; lists and forms; client-side routing; introduction to React; JSX; functional components; props and state; hooks; event handling in React; controlled components; basic state management; component composition; consuming REST APIs in React applications; building single-page applications; basic frontend project structure and development practices.

UNIT V

Advanced Interface Practices and Testing

Responsive and adaptive interfaces; accessibility and inclusive design; interface performance; browser rendering and optimization; code organization and maintainability; frontend security fundamentals; input validation and safe handling of user data; client-side testing; unit and component testing; end-to-end testing; debugging and browser profiling; introduction to frontend testing tools; deployment of JavaScript applications; progressive web applications; basic principles of design systems; emerging trends in modern JavaScript-based user interface development.

Text Books

Text Books

1. Marijn Haverbeke, *Eloquent JavaScript: A Modern Introduction to Programming*, 4th Edition, No Starch Press.
2. Eric Elliott, *Programming JavaScript Applications*, O'Reilly Media.

Reference Books

Reference Books

1. David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly Media.
2. Robin Wieruch, *The Road to React*, Independently Published.
3. Addy Osmani, *Learning JavaScript Design Patterns*, O'Reilly Media.

Course Outcomes

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

- Explain the fundamental principles of user interface engineering, usability, responsive design, and web accessibility.
- Develop interactive web interfaces using HTML, CSS, JavaScript, DOM manipulation, and event-driven programming.
- Implement asynchronous JavaScript applications that consume RESTful APIs and manage dynamic application data.
- Develop component-based user interfaces using JavaScript and React, including components, props, state, hooks, and basic routing.
- Apply responsive design, accessibility, performance optimization, debugging, and testing practices to develop maintainable user interfaces.

Internet of Things

Course Code	As per the choice of semester and paper
Course Name	Internet of Things
Credits	3L:0T: 0 P (3 Credits)
Pre-Requisites	Computer Networks

Course Objectives

- To understand the concept, architecture, applications, and challenges of IoT. To provide a thorough understanding of the sensors, actuators, and their role in IoT To explore the different communication protocols used in IoT for data transmission and device interaction. To explore the IoT cloud platforms, their architecture, and how IoT devices can be integrated with the cloud for data processing and analytics. To explore security issues of IoT systems and applications.

Syllabus

UNIT I

Introduction to IoT: Definition and evolution of IoT, Challenges in IoT, Elements of IoT, Cyber Physical Systems, Systems of Systems

Applications of IoT: Smart City,Smart homes, health care, smart cities, agriculture etc. IoT uses in Disaster Management

IoT Architecture: Layered architecture: Perception, Network, and Application layers, Cloud and Fog computing in IoT

UNIT II

Sensors in IoT: Types of sensors: Temperature, Pressure, Humidity, Proximity, Motion, Light, etc.Sensor characteristics: Accuracy, sensitivity, range, and power consumption. Introduction to Wireless Sensor Networks

Actuators in IoT: Types of actuators: Motors, Relays, Servo motors, Solenoids, etc., Actuator control and interfacing with microcontrollers;

IoT development boards: Arduino, ESP8266, ESP32, Raspberry Pi, operating systems

UNIT III

Communication Technologies: Wireless Communication: Wi-Fi, Bluetooth, Zigbee, LoRa, NB-IoT, 6LoWPAN

Wired Communication: Ethernet, Modbus, RS-232 etc

IoT Protocols: AMQP(Advanced Message Queuing Protocol), MQTT (Message Queuing Telemetry Transport),CoAP (Constrained Application Protocol), HTTP/HTTPS and REST APIs, XMPP (Extensible Messaging and Presence Protocol)

Low Power Communication: Concepts of Low Power Wide Area Networks (LPWAN), LoRaWAN,

UNIT IV

IoT Platforms: Introduction to popular IoT platforms: ThingSpeak, ThingsBoard, AWS IoT Core, Microsoft Azure IoT, Edge AI

Cloud Architecture for IoT: Data storage and processing in the cloud. IoT data analytics and visualization tools, Edge computing in IoT systems, Data Security and Privacy in IoT Cloud

UNIT V

IoT Security: Security challenges: Device authentication, data integrity, and confidentiality, Cryptographic algorithms, IoT security standards and protocols (e.g., OAuth, JWT), Secure firmware updates and Over-the-Air (OTA) updates, Secure Boot, Hardware Root of Trust, Trusted Platform Module (TPM), Zero Trust for IoT, Green IoT

IoT Case Studies

Reference Books

Reference Books

1. Raj Kamal, Internet of Things: Architecture and Design Principles, Mc Graw Hill India, 2017 Adrian McEwen & Hakim Cassimally – Designing the Internet of Things Arshdeep Bahga, and Vijay Madisetti, Internet of Things - A Hands-on Approach, Universities Press, First Edition, 2015 David Hanes, Gonzalo Salgueiro, and Patrick Grossetete, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, CISCO Press, 2017 Ovidiu Vermesan & Peter Friess – Internet of Things: Converging Technologies

Course Outcomes

- Students will understand the basic principles and applications of IoT. Familiarize themselves with IoT architecture and technologies Gain practical knowledge of sensors and actuators used in IoT systems. Learn how to interface sensors and actuators with microcontrollers. Understand the various communication protocols and their suitability for different IoT applications. Ability to implement and manage communication technologies in IoT devices. Learn to use IoT cloud platforms for device management, data storage, and analytics. Learn how to integrate IoT devices with cloud-based systems. Understand the security and privacy implications in IoT systems and apply mitigation techniques.

Evolutionary Algorithms

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Evolutionary Algorithms
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Data Structures, Probability and Statistics, and Basic Optimization

Course Objectives

The course aims to:

- Introduce the fundamental concepts and principles of evolutionary computation and population-based optimization.
- Develop an understanding of genetic algorithms and their fundamental operators and evolutionary processes.
- Introduce major evolutionary computation techniques including evolutionary strategies, evolutionary programming, and genetic programming.
- Enable students to formulate and solve basic optimization problems using evolutionary algorithms.
- Familiarize students with multi-objective optimization, practical applications, and limitations of evolutionary algorithms.

Syllabus

UNIT I

Fundamentals of Evolutionary Computation

Introduction to evolutionary computation; nature-inspired computation; evolutionary principles and natural selection; population-based optimization; search spaces and objective functions; fitness functions; exploration and exploitation; representation of solutions; evolutionary cycle; initialization, evaluation, selection, variation and replacement; stochastic optimization; advantages and limitations of evolutionary algorithms; applications in optimization and intelligent systems.

UNIT II

Genetic Algorithms

Introduction to Genetic Algorithms (GAs); chromosome representation; binary, real-valued and permutation representations; population initialization; fitness evaluation; selection mechanisms; roulette-wheel selection; tournament selection; rank selection; elitism; genetic operators; crossover

and mutation; single-point, multi-point and uniform crossover; mutation operators; replacement strategies; termination criteria; parameter selection; implementation of a basic genetic algorithm; applications of genetic algorithms.

UNIT III

Evolutionary Strategies, Evolutionary Programming and Genetic Programming

Evolutionary Strategies (ES); solution representation and mutation; selection and replacement strategies; (μ, λ) and $(\mu + \lambda)$ strategies; Evolutionary Programming (EP); representation and mutation-based search; comparison of Genetic Algorithms, Evolutionary Strategies and Evolutionary Programming; Genetic Programming (GP); representation using program trees; genetic operators in GP; fitness evaluation; applications of genetic programming.

UNIT IV

Evolutionary Optimization

Evolutionary algorithms for continuous and discrete optimization; constraint handling; penalty functions; multi-variable optimization; combinatorial optimization; scheduling and routing problems; feature selection; parameter optimization; hybrid evolutionary algorithms; combining evolutionary algorithms with local search; basic concepts of memetic algorithms; practical considerations in evolutionary optimization.

UNIT V

Multi-Objective Evolutionary Optimization and Applications

Single-objective versus multi-objective optimization; conflicting objectives; Pareto dominance; Pareto-optimal solutions; Pareto front; diversity and convergence; introduction to multi-objective evolutionary algorithms; NSGA-II and its basic principles; multi-objective fitness and selection; performance measures; applications in engineering design, scheduling, machine learning, networking and resource optimization; limitations and emerging directions in evolutionary computation.

Text Books

Text Books

1. David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
2. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd Edition, Springer.

Reference Books

Reference Books

1. Thomas Bäck, David B. Fogel and Zbigniew Michalewicz, *Handbook of Evolutionary Computation*, Oxford University Press.
2. Kenneth A. De Jong, *Evolutionary Computation: A Unified Approach*, MIT Press.
3. Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd Edition, Springer.

Course Outcomes

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

- Explain the fundamental principles, terminology, and evolutionary mechanisms underlying evolutionary algorithms.
- Design and implement basic genetic algorithms using appropriate representations, selection mechanisms, and genetic operators.
- Explain the principles of evolutionary strategies, evolutionary programming, and genetic programming.
- Apply evolutionary algorithms to basic continuous, discrete, combinatorial, and constrained optimization problems.
- Explain the fundamental concepts of multi-objective evolutionary optimization and analyse the applications and limitations of evolutionary algorithms.

Formal Methods for System Verification

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Formal Methods for System Verification
Category of Course	Professional Elective Courses
Credits	3L:0T: 0P C:3
Pre-Requisites	General knowledge in discrete mathematics, and C/C++ programming skill.

Course Objectives

- Introduce students to the mathematically proving formally specified properties of computer systems. Students will be able to learn on theoretical aspects of specification formalisms and algorithm verifications. Students should be able to understand mathematically-based techniques for the specification, development, and verification of software and hardware systems. Students will acquire skill in using language for model description and specification of model behaviors in modeling and verification of event-driven systems.

Syllabus

UNIT I

Introduction to the formal methods and modeling systems

Introduction: The need for formal methods, Motivation for formal verification, Hardware and software verification, simple verification examples. Model Checking and Theorem Proving, Spectrum of logic with examples

UNIT II

Logical formalism

Propositional logic: Declarative sentences, Natural deduction, Propositional logic as a formal language, Syntax and Semantics of propositional logic.

Predicate logic: Syntax, Semantis and examples, First order logic- Axioms, decidability of first order logic, higher order logic, normal forms- NNF, CNF, DNF, conversion of normal forms, Tsetin transformation.

UNIT III

SAT Solving: Introduction conflict clause, Backtrack, analyze clause, introduction SAT solver, different techniques for SAT solving, like implication graph, Backtrack, CDCL, DPLL algorithm, decision heuristics, DIMACS format, preprocessing of SAT formula, case study on SAT based modeling.

BDD, ROBDD, reduction rules, variable ordering in BDD, Formal property verification (FPV), introduction to Model Checking, different properties of FPV

UNIT IV

CTL model checking- Syntax and semantic, encoding examples.

LTL model checking- Syntax and semantic, encoding examples.

CTL and LTL model checking algorithms, correctness of model checking algorithm

UNIT V

Model checking and Automata Theory

Introduction: Automata on finite and infinite words, Model checking using automata, Checking emptiness, Translating LTL into automata, On-the-Fly model checking.

Text Books

Text Books

1. Logic in Computer Science: Modelling and Reasoning about Systems, 2nd Edition, M. Huth and M. Ryan, published by Cambridge University Press, Copyright © 2011 (Reprinted 2007, 2010, 2011). Model Checking, E. M. Clarke, O. Grumberg and D. Peled, MIT Press, 1999. Principles of Model Checking, Christel Baier and Joost-Pieter Katoen, MIT Press, May 2008. Daniel Kroening and Ofer Strichman: Decision Procedures - An Algorithmic Point of View, Springer, 2008

Reference Books

Reference Books

1. Higher Order Logic and Hardware Verification, T. F. Melham, published by Cambridge University Press, Print publication year: 1993, online publication date: January, 2010. Algorithm Design, 1st Edition, Jon Kleinberg, and Eva Tardos, published by Pearson Education Limited, Copyright © 2014. T. F. Melham, Higher Order Logic and Hardware Verification, Cambridge University Press, 1993. The Temporal Logic of Reactive and Concurrent System Specification, Z. Manna and A. Pnueli, Springer Verlag, 1992.

Course Outcomes

- Students will be able to understand formal methods which are applying for Hardware and Software verifications. Students will be able to write the formal proofs based on the propositional logic, predicate logic, and temporal logic to verify the hardware circuits and program verifications. Students will be capable of writing the formal properties and specifications in computation tree logic (CTL), linear-time temporal logic (LTL). Students will be able to verify the systems using CTL and LTL model checking. Students will be able to construct and use Binary Decision Diagrams (BDDs) in symbolic model checking. Students will be able to understand the model checking using automata and also able to translating LTL into automata. Students will be able to learn the verification tools: SMV, PVS.

Full Stack Development

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Full Stack Development
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Programming Fundamentals, Data Structures and Database Management Systems

Course Objectives

The course aims to:

- Introduce the fundamental concepts and architecture of full stack web application development.
- Develop proficiency in designing responsive and interactive client-side web interfaces.
- Introduce server-side application development, routing, APIs, authentication, and backend services.
- Develop an understanding of database integration and communication between frontend, backend, and data layers.
- Familiarize students with testing, security, version control, and deployment practices for modern full stack applications.

Syllabus

UNIT I

Web Application Fundamentals and Frontend Development

Introduction to full stack development; web application architecture; client–server model; HTTP and HTTPS; request–response cycle; URLs and web resources; HTML5 and semantic HTML; forms and validation; CSS fundamentals; responsive web design; JavaScript fundamentals; DOM manipulation; events; asynchronous JavaScript; JSON; browser storage; introduction to frontend development tools and package management.

UNIT II

Modern Frontend Development

Component-based user interface development; introduction to frontend frameworks and libraries; React fundamentals; JSX; components, properties and state; event handling; hooks; conditional rendering; lists and forms; component composition; client-side routing; consuming REST APIs; application state management; reusable components; responsive and accessible interfaces; frontend project structure and development practices.

UNIT III

Backend Development and RESTful Services

Introduction to server-side programming; Node.js fundamentals; npm and package management; modules; asynchronous programming; Express.js; server and routing; middleware; request and response handling; RESTful API design; HTTP methods and status codes; JSON-based communication; input validation; error handling; environment configuration; authentication and authorization; cookies, sessions and tokens; basic backend application structure.

UNIT IV

Database Integration and Full Stack Applications

Role of databases in web applications; relational and NoSQL databases; database schema and data modelling; SQL fundamentals; CRUD operations; database connectivity from backend applications; Object–Relational Mapping (ORM) and database abstraction; MongoDB fundamentals; data validation and indexing; connecting frontend, backend and database layers; API-based data exchange; complete request flow in a full stack application; basic transaction and data consistency concepts.

UNIT V

Testing, Security and Deployment

Testing full stack applications; frontend and backend testing; API testing; unit and integration testing; debugging; version control using Git; application configuration and environment variables; web application security fundamentals; input validation; authentication and authorization; common web vulnerabilities; Cross-Site Scripting (XSS); Cross-Site Request Forgery (CSRF); secure API development; deployment of full stack applications; cloud and platform-based deployment; build and release processes; application monitoring; introduction to containerization and modern full stack development practices.

Text Books

Text Books

1. Marijn Haverbeke, *Eloquent JavaScript: A Modern Introduction to Programming*, 4th Edition, No Starch Press.
2. Robin Wieruch, *The Road to React*, Independently Published.
3. Ethan Brown, *Web Development with Node and Express*, 2nd Edition, O'Reilly Media.

Reference Books

Reference Books

1. David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly Media.
2. Mario Casciaro and Luciano Mammino, *Node.js Design Patterns*, 3rd Edition, Packt Publishing.
3. Adam Freeman, *Pro React 18*, Apress.
4. Martin Kleppmann, *Designing Data-Intensive Applications*, O'Reilly Media.

Course Outcomes

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

- Explain the architecture and fundamental components of modern full stack web applications.
- Develop responsive and interactive frontend applications using HTML, CSS, JavaScript, and component-based frameworks.
- Develop backend services and RESTful APIs using server-side technologies and implement basic authentication and authorization.
- Integrate relational and NoSQL databases with backend applications and implement complete data-driven web applications.
- Apply basic testing, security, version control, and deployment practices to develop and deploy maintainable full stack applications.

Game Development

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Game Development
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Programming Fundamentals, Object-Oriented Programming and Basic Computer Graphics

Course Objectives

The course aims to:

- Introduce the fundamental concepts, components, and development processes involved in digital game development.
- Develop practical skills in game programming using C# and the Unity game engine.
- Introduce the fundamentals of 2D and 3D game development, animation, physics, and interactive game environments.
- Familiarize students with basic game artificial intelligence, navigation, level design, and player interaction.
- Develop an understanding of game testing, debugging, optimization, and deployment using modern game development tools, including open-source alternatives.

Syllabus

UNIT I

Fundamentals of Game Development and Unity

Introduction to digital games and game development; game genres and platforms; game development lifecycle; game mechanics, rules and objectives; game states and player interaction; introduction to game engines; Unity Engine and its architecture; Unity Editor and project structure; scenes, GameObjects and Components; Prefabs and Assets; Transform component; Tags and Layers; Unity scripting using C#; variables, methods, classes and objects; event-driven programming; input handling; basic game logic; introduction to Godot Engine as an open-source/free-software alternative for game development.

UNIT II

2D Game Development Using Unity

2D game development workflow; sprites and sprite sheets; 2D GameObjects; Sprite Renderer; cameras and transformations; animation and Animator Controller; user input and controls; collision detection and physics; Rigidbody and Collider components; triggers; tile-based level construction; audio and sound effects; user interface elements; scene management; collectibles, scoring and basic game mechanics; development of a simple 2D game using Unity and C#.

UNIT III

3D Game Development and Physics

Fundamentals of 3D game development; coordinate systems and transformations; 3D GameObjects; meshes and materials; cameras and lighting; textures and shaders; terrain and environment design; 3D animation; Animator Controller; physics bodies and collision shapes; forces, gravity and motion; Rigidbody and Character Controller; ray casting; particle systems; basic physics-based interactions; development of a simple interactive 3D game environment using Unity.

UNIT IV

Game AI, Navigation and Interaction

Introduction to artificial intelligence in games; non-player characters (NPCs); rule-based behaviour; finite state machines; decision-making; navigation and pathfinding; Unity Navigation and NavMesh; A* pathfinding; steering and movement behaviours; enemy and NPC behaviour; basic game difficulty and balancing; player interaction and feedback; introduction to procedural content generation; implementation of basic game AI using C# and Unity.

UNIT V

Game Testing, Optimization and Deployment

Game project organization and asset management; debugging and error handling; game testing and playtesting; performance profiling; frame rate and resource management; optimization of graphics, scripts and physics; memory management; version control for game projects; Unity build and deployment workflow; exporting games for desktop, web and mobile platforms; basic multiplayer game concepts; accessibility and responsible game design; introduction to Godot-based game development workflows; comparison of Unity and Godot development approaches; development and deployment of a complete small game project.

Primary Development Framework

- **Primary Framework:** Unity Engine with C#.
- **Open-Source/Free-Software Alternative:** Godot Engine with GDScript or C#.

Text Books

Text Books

1. Jeremy Gibson Bond, *Introduction to Game Design, Prototyping, and Development*, 2nd Edition, Addison-Wesley.
2. Ernest Adams, *Fundamentals of Game Design*, 3rd Edition, New Riders.
3. Joseph Hocking, *Unity in Action: Multiplatform Game Development in C#*, 3rd Edition, Manning Publications.

Reference Books

Reference Books

1. Jason Gregory, *Game Engine Architecture*, 3rd Edition, CRC Press.

2. Ian Millington, *Artificial Intelligence for Games*, 3rd Edition, CRC Press.
3. Robert Nystrom, *Game Programming Patterns*, Genever Benning.
4. David H. Eberly, *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*, 2nd Edition, CRC Press.

Course Outcomes

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

- Explain the fundamental concepts, components, genres, and development processes of digital games and game engines.
- Develop basic 2D games using Unity, C#, scenes, GameObjects, components, sprites, animation, input handling, and collision detection.
- Develop basic interactive 3D game environments using Unity and apply fundamental concepts of cameras, lighting, animation, materials, and physics.
- Implement basic game AI techniques including finite state machines, navigation, pathfinding, and NPC behaviour.
- Apply testing, debugging, optimization, and deployment techniques to develop and distribute game applications, and explain the role of Godot as an open-source/free-software alternative to Unity.

Human Computer Interaction

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Human Computer Interaction
Category of Course	Professional Elective Courses
Credits	3L:0T:0P C:3
Pre-Requisites	Programming Fundamentals and Basic Computer Systems

Course Objectives

The course aims to:

- Introduce the fundamental concepts, theories, and interdisciplinary foundations of Human Computer Interaction.
- Develop an understanding of human perception, cognition, behaviour, and motor abilities relevant to interactive computing.
- Introduce fundamental models and theories for understanding human–computer interaction and interactive system behaviour.
- Familiarize students with basic qualitative and quantitative research methods used in HCI research and evaluation.
- Introduce contemporary research directions in HCI, including ubiquitous, multimodal, immersive, and human–AI interaction.

Syllabus

UNIT I

Foundations of Human Computer Interaction

Introduction to Human Computer Interaction; history and evolution of HCI; scope and interdisciplinary nature of HCI; relationship between computer science, psychology, cognitive science, design and social sciences; human, computer and interaction components; characteristics of interactive systems; usability and user experience; effectiveness, efficiency and satisfaction; human-centred computing; interaction as a research problem; ethical considerations in HCI research.

UNIT II

Human Factors and Interaction Models

Human perception and sensory systems; visual, auditory and haptic interaction; attention and memory; learning and problem solving; mental models; decision making; human motor behaviour; Fitts's Law; Hick–Hyman Law; cognitive load; human error; affordances and feedback; conceptual models; interaction frameworks; Norman's interaction model; distributed cognition; activity theory; understanding and modelling human interaction with computing systems.

UNIT III

Interaction and User Studies

Interaction styles and modalities; graphical, tangible, touch, gesture, voice and conversational interaction; direct manipulation; mobile and ubiquitous interaction; multimodal interaction; user-centred and human-centred approaches; research questions and hypotheses in HCI; participant selection; experimental design; observational studies; interviews; questionnaires; contextual inquiry; think-aloud studies; qualitative and quantitative data collection; ethical considerations and informed consent in user studies.

UNIT IV

Empirical Evaluation and HCI Research Methods

Usability evaluation and empirical research; experimental and quasi- experimental studies; controlled experiments; within-subject and between-subject designs; usability metrics; task completion time, error rate and satisfaction; qualitative data analysis; descriptive statistics; basic statistical testing for HCI studies; heuristic evaluation; cognitive walkthrough; A/B testing; reliability and validity; threats to validity; reproducibility and research reporting; interpreting and presenting HCI research results.

UNIT V

Emerging Research in Human Computer Interaction

Accessibility and inclusive interaction; assistive technologies; ubiquitous and pervasive computing; wearable and tangible interfaces; virtual and augmented reality; human–robot interaction; affective computing; brain–computer interaction; social computing; collaborative interaction; conversational interfaces; human–AI interaction; explainability, trust and reliance in AI systems; privacy and human autonomy; emerging research problems and directions in Human Computer Interaction.

Text Books

Text Books

1. Alan Dix, Janet Finlay, Gregory D. Abowd and Russell Beale, *Human–Computer Interaction*, 3rd Edition, Pearson.
2. Jenny Preece, Yvonne Rogers and Helen Sharp, *Interaction Design: Beyond Human–Computer Interaction*, 5th Edition, Wiley.

Reference Books

Reference Books

1. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist and Nicholas Diakopoulos, *Designing the User Interface: Strategies for Effective Human–Computer Interaction*, 6th Edition, Pearson.
2. Alan Cooper, Robert Reimann, David Cronin and Christopher Noessel, *About Face: The Essentials of Interaction Design*, 4th Edition, Wiley.
3. Jonathan Lazar, Jinjuan Heidi Feng and Harry Hochheiser, *Research Methods in Human–*

Computer Interaction, 2nd Edition, Morgan Kaufmann.

4. Andrew Sears and Julie A. Jacko, *The Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, 3rd Edition, CRC Press.

Course Outcomes

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

- Explain the fundamental concepts, theories, and interdisciplinary foundations of Human Computer Interaction.
- Analyse the influence of human perception, cognition, memory, motor behaviour, and errors on interaction with computing systems.
- Apply fundamental HCI models and principles to analyse human interaction with interactive systems.
- Design basic HCI studies and apply qualitative and quantitative methods to collect, analyse, and interpret user interaction data.
- Evaluate emerging HCI research areas and identify research challenges related to accessibility, ubiquitous computing, immersive interaction, human–AI interaction, and emerging technologies.

Information Retrieval

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Information Retrieval
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Any Programming Language (Preferably java) Probability and Linear algebra

Course Objectives

- To learn and study algorithms which will enable to design, and implement modern information retrieval systems.
- To investigate search evaluation, retrieval feedback, search log mining, and applications in web information management.

Syllabus

UNIT I

Introduction

Introduction, History Of IR, Components Of IR, Issues, Open Source

Search Engine Frameworks, The Impact Of The Web On IR, IR Versus Web Search, The Basic Building Blocks Of A Modern Search Engine System, Including Web Crawler, Basic Text Analysis Techniques.

UNIT II

Models of Information Retrieval

Inverted Index, Query Processing, Search Result Interface. Boolean And Vector-Space Retrieval Models, Term Weighting, TF-IDF

Weighting, Cosine Similarity, Preprocessing, Efficient Processing With Sparse Vectors, Language Model Based IR, Probabilistic IR, Latent Semantic Indexing, Relevance Feedback And Query Expansion.

UNIT III

Text Mining

Information Filtering; Organization And Relevance Feedback, Text Mining, Text Classification And Clustering.

Categorization Algorithms

Naive Bayes; Decision Trees; And Nearest Neighbor.

Clustering Algorithms

Agglomerative Clustering; KMeans; Expectation Maximization (EM).

UNIT IV

Link Analysis

Link Analysis, Hubs And Authorities, Page Rank And HITS Algorithms, Searching And Ranking, Relevance Scoring And Ranking For Web.

UNIT V

Similarity and evaluation measures

Evaluation Measures, Similarity And Distance Measures, Snippet Generation, Summarization, Question Answering, Cross-Lingual Retrieval, Hadoop& Map Reduce And Modern Search Applications.

Text Books

Text Books

1. C.Manning,P.Raghavan,and H.Schutze,Introduction to Information Retrieval,Cambridge University Press, 2008.
2. Bruce Croft, Donald Metzler and Trevor Strohman,Search Engines: Information Retrieval in Practice, 11st Edition, Addison Wesley, 2009.
3. MarkLevene,An Introduction to Search Engines and Web Navigation,2nd Edition Wiley.

Reference Books

Reference Books

1. OphirFrieder“InformationRetrieval:Algorithms and Heuristics:The Information Retrieval Series”, nd 2nd Edition,Springer, 2004.
2. ManuKonchady,“Building Search Applications:Lucene,LingPipe”,and First Edition,GateMustru Publishing, 2008.

Course Outcomes

- Recognize underlying technologies of modern information retrieval system.
- Obtain hands-on experience by using existing information retrieval toolkits to set up your own search engines and improving their search accuracy.
- Helps in gaining in-depth understanding of the methods like document text-mining techniques, page-rank etc. and develop your own idea for new solutions for different verticals.

OPEN ELECTIVES

Exploring Knowledge Beyond Disciplinary Boundaries

Interdisciplinary courses designed to broaden perspectives and develop diverse competencies.

Fundamentals of Machine Learning

Course Code	CSE603
Course Name	Fundamentals of Machine Learning
Category of Course	Professional Core Courses (PC)
Credits	3L: 0T: 0P C:3
Pre-Requisites	Probability and Linear Algebra; Basic programming skills preferably in Python

Course Objectives

The students will be able to:

- Understand the basic concepts of machine learning and algorithms.
- Understand concepts related to supervised and unsupervised learning.
- Understand the basic concepts of deep learning and learning theory.
- Understand issues related to the application of machine learning algorithms.

Syllabus

UNIT I

Fundamentals

Introduction, Different Types of Learning, Hypothesis Space and Inductive Bias, Evaluation and Cross-Validation, Linear Regression, Introduction to Decision Trees, Learning Decision Tree, Overfitting.

UNIT II

Supervised Learning

Supervised learning setup, LMS, Logistic Regression, Perceptron, Exponential Family, Generative Learning Algorithms, Gaussian Discriminant Analysis, Naive Bayes, Support Vector Machines.

UNIT III

Learning Theory

Model Selection and Feature Selection, Ensemble Methods: Bagging, Boosting, Evaluating and Debugging Learning Algorithms; Bias/Variance Tradeoff, Union and Chernoff/Hoeffding Bounds, VC Dimension, Worst-Case (Online) Learning.

UNIT IV

Unsupervised Learning

Clustering: K-means, EM, Mixture of Gaussians, Factor Analysis, PCA (Principal Components Analysis), ICA (Independent Components Analysis).

UNIT V

Reinforcement Learning and Control

MDPs, Bellman Equations, Value Iteration and Policy Iteration, Linear Quadratic Regulation (LQR), LQG, Q-Learning, Value Function Approximation, Policy Search, Reinforce, POMDPs.

Text Books

Text Books

1. Ethem Alpaydin, *Introduction to Machine Learning*, 3rd Edition, PHI, 2015.
2. Tom M. Mitchell, *Machine Learning*, McGraw Hill Education, First Edition, 2017.

Reference Books

Reference Books

1. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition, 2009.
2. Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer, 2011 Edition.

Course Outcomes

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

- For a given problem, analyze and implement the solution using Linear Regression.
- For a given problem, analyze and implement the solution using Logistic Regression, Decision Trees, and K-Nearest Neighbor.
- Apply Bayesian learning and the Naive Bayes algorithm to appropriate learning problems.
- Apply Support Vector Machines and kernels to solve classification problems.
- Apply Neural Networks to determine and justify the correctness of a given solution.
- Analyze hypothesis space, overfitting, bias, and variance.
- Analyze trade-offs between representational power and learnability.
- Apply evaluation strategies, cross-validation, and feature reduction methods.

Data Analytics

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Data Analytics
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Data Structure, Statistics, Data Mining

Course Objectives

The Student should be made to:

- Be exposed to big data.
- Learn the different ways of Data Analysis.
- Be familiar with data streams.
- Learn the mining and clustering.
- Be familiar with the visualization.

Syllabus

UNIT I

Introduction to Big data

Introduction to Big Data Platform – Challenges of conventional systems - Web data – Evolution of Analytic scalability, analytic processes and tools, Analysis vs. reporting - Modern data analytic tools,

Statistical concepts

Sampling distributions, resampling, statistical inference, and prediction error.

UNIT II

Data analysis

Regression modeling, Multivariate analysis, Bayesian modeling, inference and Bayesian networks, and Support vector and kernel methods, Analysis of time series: linear systems analysis, nonlinear dynamics - Rule induction - Neural networks: learning and generalization, competitive learning, principal component analysis and neural networks.

Fuzzy logic

Extracting fuzzy models from data, fuzzy decision trees, Stochastic search methods.

UNIT III

Mining data streams: Introduction to Streams Concepts – Stream data model and architecture - Stream Computing, Sampling data in a stream – Filtering streams – Counting distinct elements in a stream – Estimating moments – Counting oneness in a window – Decaying window - Real-time Analytics Platform (RTAP) applications - case studies - real time sentiment analysis, stock market predictions.

UNIT IV

Frequent item sets and clustering Mining Frequent item sets - Market based model – Apriori Algorithm – Handling large data sets in Main memory – Limited Pass algorithm – Counting frequent item sets in a stream – Clustering Techniques – Hierarchical – K- Means – Clustering high dimensional data – CLIQUE and PROCLUS – Frequent pattern based clustering methods – Clustering in non-Euclidean space – Clustering for streams and Parallelism.

UNIT V

Frameworks and visualization

MapReduce – Hadoop, Hive, MapR – Sharding – NoSQL Databases - S3

- Hadoop Distributed file systems – Visualizations - Visual data analysis techniques, interaction techniques; Systems and applications.

Text Books

Text Books

1. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, ”Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today’s Businesses
2. Montgomery, Douglas C. and Runger, George C. (2014) Applied Statistics and Probability for Engineers, 6th edition, John Wiley & Sons, Inc (ISBN- 978-1118539712) Reference books:
3. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
4. Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 2012.
5. P. J. Sadalage and M. Fowler, ”NoSQL Distilled: A Brief Guide to the Emerging World of
6. Tom White, ”Hadoop: The Definitive Guide”, Third Edition, O’Reilly, 2012.
7. Eric Sammer, ”Hadoop Operations”, O’Reilly, 2012.

Course Outcomes

- After completion of course, students would be:
- Demonstrate understanding of hypotheses testing for samples to solve engineering problems.
- Perform linear and multiple linear regression analyses.
- Demonstrate ability to design and analysis of single-factor experiments.
- Demonstrate ability to do design of experiments with several factors.

- Describe big data and use cases from selected business domains.
- Explain NoSQL big data management.
- Install, configure, and run Hadoop and HDFS.
- Perform map-reduce analytics using Hadoop.
- Use Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics.

Data Science

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Data Science
Category of Course	Open Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic knowledge of mathematics and programming is desirable

Course Objectives

The students will be able to:

- Understand the fundamental concepts, processes, and applications of Data Science.
- Develop the ability to collect, organize, clean, and prepare data for analysis.
- Apply basic statistical and exploratory techniques to derive meaningful insights from data.
- Understand fundamental machine learning techniques used for prediction and classification.
- Develop awareness of data visualization, ethical issues, and real-world applications of Data Science.

Syllabus

UNIT I

Introduction to Data Science

Introduction to Data Science and its importance, Data Science life cycle, types and sources of data, structured and unstructured data, data collection and data acquisition, roles of Data Scientists and Data Analysts, data-driven decision making, and applications of Data Science in healthcare, business, education, agriculture, finance, and other domains.

UNIT II

Data Preparation and Exploratory Data Analysis

Data preprocessing and cleaning, handling missing values and duplicate records, identifying and treating outliers, data transformation and normalization, basic data integration, descriptive statistics, measures of central tendency and dispersion, exploratory data analysis, correlation, and introduction to data analysis using spreadsheets or programming tools.

UNIT III

Data Visualization and Statistical Analysis

Principles of effective data visualization, charts and graphs for data analysis, bar charts, line charts, histograms, scatter plots, box plots, and heat maps, visual interpretation of data, probability concepts, random variables, common probability distributions, sampling, hypothesis testing, confidence intervals, and interpretation of statistical results.

UNIT IV

Introduction to Machine Learning

Introduction to Machine Learning and its relationship with Data Science, supervised and unsupervised learning, training and testing data, feature selection and feature engineering, regression and classification, Linear Regression, Logistic Regression, Decision Trees, K-Nearest Neighbors, clustering using K-means, model evaluation, accuracy, precision, recall, and overfitting and underfitting.

UNIT V

Data Science Applications and Responsible Data Practices

End-to-end Data Science workflow, introduction to big data and cloud-based data analysis, real-world Data Science case studies, introduction to recommendation and predictive systems, data privacy and security, ethical issues in Data Science, bias and fairness in data-driven systems, responsible use of data, and emerging trends in Data Science and Artificial Intelligence.

Text Books

Text Books

1. Foster Provost and Tom Fawcett, *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*, O'Reilly Media, 2013.
2. Joel Grus, *Data Science from Scratch: First Principles with Python*, O'Reilly Media, 2nd Edition, 2019.
3. Cathy O'Neil and Rachel Schutt, *Doing Data Science: Straight Talk from the Frontline*, O'Reilly Media, 2013.

Reference Books

Reference Books

1. Wes McKinney, *Python for Data Analysis*, O'Reilly Media, 3rd Edition, 2022.
2. Jake VanderPlas, *Python Data Science Handbook*, O'Reilly Media, 2016.
3. Peter Bruce, Andrew Bruce, and Peter Gedeck, *Practical Statistics for Data Scientists*, O'Reilly Media, 2nd Edition, 2020.

Course Outcomes

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

- Explain the fundamental concepts, processes, and applications of Data Science.

- Collect, organize, clean, and preprocess data for analysis.
- Apply exploratory, statistical, and visualization techniques to interpret data and derive meaningful insights.
- Apply fundamental machine learning techniques to basic prediction, classification, and clustering problems.
- Analyze real-world Data Science applications while considering ethical, privacy, security, bias, and responsible data practices.

MATLAB

Course Code	Code will be given as per choice in particular semester and paper
Course Name	MATLAB
Category of Course	Open Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic Programming Knowledge

Course Objectives

The students will be able to:

- Understand the MATLAB Desktop, Command Window, and Graph Window.
- Perform simple and complex calculations using MATLAB.
- Carry out numerical computations and analyses using MATLAB.
- Understand the mathematical concepts upon which numerical methods rely.
- Competently use the MATLAB programming environment.
- Understand the tools that are essential for solving engineering problems.

Syllabus

UNIT I

Introduction to MATLAB

MATLAB Interactive Sessions, Computing with MATLAB, Variables, Arrays, Functions, and Files.

UNIT II

Programming Techniques

Program Design and Development, Relational Operators and Logical Variables, Logical Operators and Functions, Conditional Statements, Loops, The Switch Structure, Debugging MATLAB Programs.

UNIT III

Plotting

XY-Plotting Functions, Subplots and Overlay Plots, Special Plot Types, Interactive Plotting, Function Discovery, Regression, 3-D Plots.

UNIT IV

Probability and Statistics

Interpolation, Statistics, Histogram and Probability, The Normal Distribution, Random Number Generation.

UNIT V

Symbolic Processing with MATLAB

Symbolic Expressions and Algebra, Algebraic and Transcendental Equations, Calculus, Symbolic Linear Algebra.

Text Books

Text Books

1. William J. Palm III, *Introduction to MATLAB 7 for Engineers*, McGraw Hill, 2005.

Reference Books

Reference Books

1. S. J. Chapman, *MATLAB Programming for Engineers*, Thomson, 2004.
2. J. Cooper, *A MATLAB Companion for Multivariable Calculus*, Academic Press, 2001.

Course Outcomes

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

- Use MATLAB for interactive computations.
- Demonstrate familiarity with memory and file management in MATLAB.
- Generate plots and export graphical results for analysis and presentation.

Python Programming

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Python Programming
Category of Course	Open Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic knowledge of computer fundamentals

Course Objectives

The students will be able to:

- Understand the fundamental concepts, syntax, and programming constructs of Python.
- Develop Python programs using variables, expressions, decision-making statements, and loops.
- Apply functions, strings, lists, tuples, dictionaries, and sets to solve computational problems.
- Understand file handling, exception handling, and modular programming in Python.
- Develop basic Python applications using object-oriented programming and standard libraries.

Syllabus

UNIT I

Introduction to Python and Basic Programming

Introduction to Python and its applications, features of Python, Python programming environment, interactive and script modes, tokens, keywords, identifiers, variables, data types, input and output operations, operators and expressions, type conversion, comments, and basic program development.

UNIT II

Decision Making and Looping

Conditional statements: `if`, `if-else`, nested `if`, and `elif`. Iterative statements: `for` and `while` loops. Nested loops, `break`, `continue`, and `pass` statements. Introduction to problem solving and development of Python programs using decision-making and looping constructs.

UNIT III

Functions and Python Data Structures

Functions: defining and calling functions, function arguments, return values, scope of variables, recursion, and introduction to modules. Strings and string operations. Lists, tuples, sets, and dictionaries; creation, traversal, updating, and commonly used operations on Python data structures.

UNIT IV

File Handling and Exception Handling

Introduction to files, opening and closing files, file modes, reading and writing text files, file operations, and basic data processing using files. Exception handling using `try`, `except`, `else`, and `finally`. Introduction to user-defined modules and packages.

UNIT V

Object-Oriented Programming and Python Applications

Fundamentals of object-oriented programming, classes and objects, constructors, instance variables and methods, inheritance, and polymorphism. Introduction to selected Python standard libraries. Development of simple applications involving data processing, automation, and problem solving.

Text Books

Text Books

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, 2nd Edition.
2. Allen B. Downey, *Think Python: How to Think Like a Computer Scientist*, O'Reilly Media, 2nd Edition, 2015.

Reference Books

Reference Books

1. Mark Lutz, *Learning Python*, O'Reilly Media, 5th Edition, 2013.
2. John V. Guttag, *Introduction to Computation and Programming Using Python*, MIT Press, 3rd Edition, 2021.
3. Al Sweigart, *Automate the Boring Stuff with Python*, No Starch Press, 2nd Edition, 2019.

Course Outcomes

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

- Explain the fundamental concepts, syntax, data types, operators, and programming constructs of Python.
- Develop Python programs using conditional statements, loops, and functions to solve basic computational problems.
- Apply appropriate Python data structures and string operations for storing and processing data.
- Implement file handling, exception handling, modules, and packages in Python programs.
- Develop simple object-oriented and application-oriented Python programs for real-world problem solving.

Technical Writing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Technical Writing
Category of Course	Open Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic proficiency in English language and computer fundamentals

Course Objectives

The students will be able to:

- Understand the principles and importance of effective technical and academic writing.
- Develop skills to plan, organize, write, revise, and present technical documents.
- Apply appropriate language, structure, formatting, and visual elements in technical communication.
- Understand the fundamentals of LaTeX and use it for preparing structured technical and academic documents.
- Develop professionally formatted reports and research-oriented documents using technical writing principles and LaTeX.

Syllabus

UNIT I

Introduction to Technical Writing and LaTeX

Nature, purpose, scope, and importance of technical writing; characteristics of effective technical communication; audience, purpose, and context; clarity, conciseness, accuracy, completeness, and coherence; stages of the writing process: planning, drafting, revising, editing, and proofreading. Introduction to document preparation systems; introduction to LaTeX, its features and applications in technical and academic writing; LaTeX installation and online editors; basic structure of a LaTeX document; document classes and basic document syntax; compiling a LaTeX document.

UNIT II

Technical Writing Process and Basic LaTeX Formatting

Sentence and paragraph construction; logical organization and flow of ideas; effective word choice; formal technical style; headings, subheadings, lists, and document organization; introduction, body, and conclusion; grammar, punctuation, and common errors in technical writing. Creating and formatting documents in LaTeX; text formatting, paragraphs, sections, subsections, lists, special characters, line and page breaks; page layout, margins, headers, footers, and title pages; use of basic packages and document customization.

UNIT III

Technical Documents and LaTeX Elements

Technical definitions, descriptions, instructions, and process explanations; preparation of professional documents such as e-mails, memoranda, notices, and letters; executive summaries and documentation practices. Preparation of tables and figures in LaTeX; captions, labels, and cross-referencing; inclusion and positioning of graphics; mathematical expressions and equations; matrices, symbols, and aligned equations; introduction to commonly used packages for tables, figures, and mathematical typesetting.

UNIT IV

Technical Reports, Research Writing, and Referencing

Purpose and types of technical reports; structure and organization of technical and project reports; title, abstract, introduction, literature review, methodology, results, discussion, conclusion, and recommendations; basics of research writing; paraphrasing, summarizing, and quotation; plagiarism and academic integrity; citation and referencing practices. Preparation of technical and research documents in LaTeX; citations and bibliographies; introduction to BibTeX; creating and managing reference databases; citation commands and bibliography styles; cross-references and automated document numbering.

UNIT V

Advanced LaTeX and Integrated Technical Document Preparation

Advanced document structuring and customization; multi-file documents and reusable commands; theorem-like environments and specialized mathematical formatting; advanced tables and figures; hyperlinks and internal navigation; introduction to presentation preparation using Beamer; preparation of slides with text, lists, figures, tables, and mathematical content. Integration of technical writing principles with LaTeX document preparation; planning, writing, formatting, reviewing, and finalizing a complete technical report or academic document; contemporary tools and practices for technical and scientific documentation.

Text Books

Text Books

1. Meenakshi Raman and Sangeeta Sharma, *Technical Communication: Principles and Practice*, Oxford University Press.
2. Mike Markel and Stuart A. Selber, *Technical Communication*, Bedford/St. Martin's.
3. Leslie Lamport, *LaTeX: A Document Preparation System*, Addison-Wesley.

Reference Books

Reference Books

1. Gerald J. Alred, Charles T. Brusaw, and Walter E. Oliu, *Handbook of Technical Writing*, Bedford/St. Martin's.
2. Sharon J. Gerson and Steven M. Gerson, *Technical Writing: Process and Product*, Pearson.
3. Frank Mittelbach and Michel Goossens, *The LaTeX Companion*, Addison-Wesley.

4. Helmut Kopka and Patrick W. Daly, *A Guide to LaTeX*, Addison-Wesley.

Course Outcomes

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

- Explain the principles, processes, and characteristics of effective technical and academic writing.
- Prepare well-organized technical documents using appropriate language, structure, and formatting.
- Create structured LaTeX documents using document classes, sections, lists, formatting commands, and relevant packages.
- Prepare technical reports containing tables, figures, mathematical expressions, citations, references, and bibliographies using LaTeX.
- Develop and present complete professionally formatted technical or academic documents by integrating technical writing principles with LaTeX tools.

Soft Computing

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Soft Computing
Category of Course	Professional Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic knowledge of mathematics, probability, and programming

Course Objectives

The students will be able to:

- Understand the fundamental concepts, characteristics, and applications of Soft Computing.
- Develop an understanding of fuzzy sets, fuzzy logic, and fuzzy inference systems.
- Understand artificial neural networks and their learning algorithms for solving computational problems.
- Apply evolutionary and swarm intelligence techniques for optimization and search problems.
- Analyze and design hybrid Soft Computing systems for real-world applications.

Syllabus

UNIT I

Introduction to Soft Computing and Fuzzy Logic

Introduction to Soft Computing, characteristics and components of Soft Computing, differences between conventional computing and Soft Computing, applications of Soft Computing. Introduction to fuzzy sets and fuzzy logic; crisp sets versus fuzzy sets; membership functions and their types; operations on fuzzy sets; linguistic variables; fuzzy relations; fuzzy propositions and rules; fuzzy inference systems; fuzzification and defuzzification methods.

UNIT II

Fuzzy Systems and Applications

Fuzzy reasoning and approximate reasoning; Mamdani and Sugeno fuzzy inference models; composition of fuzzy relations; fuzzy rule-based systems; fuzzy controllers and their structure; design of simple fuzzy systems; applications of fuzzy logic in decision making, pattern recognition, control, and engineering systems.

UNIT III

Artificial Neural Networks

Introduction to artificial neural networks; biological inspiration and artificial neuron models; neural network architectures; activation functions; supervised and unsupervised learning; Perceptron and Multilayer Perceptron; learning rules; backpropagation algorithm; Hopfield networks; Self-Organizing Maps; Radial Basis Function networks; training, testing, and performance evaluation of neural networks.

UNIT IV

Evolutionary and Swarm Intelligence Techniques

Introduction to evolutionary computation and optimization; Genetic Algorithms: representation, population initialization, fitness functions, selection, crossover, mutation, and termination criteria; applications of Genetic Algorithms. Introduction to swarm intelligence; Particle Swarm Optimization; Ant Colony Optimization; comparison of evolutionary and swarm-based optimization techniques; applications in search, optimization, scheduling, and engineering design.

UNIT V

Hybrid Soft Computing and Applications

Hybrid Soft Computing systems; neuro-fuzzy systems; Adaptive Neuro-Fuzzy Inference System (ANFIS); genetic-fuzzy systems; evolutionary neural networks; selection of appropriate Soft Computing techniques; performance evaluation and comparison of Soft Computing models. Applications in classification, prediction, optimization, pattern recognition, control systems, data analysis, image processing, and intelligent decision support systems.

Text Books

Text Books

1. S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing*, Wiley India.
2. Jyh-Shing Roger Jang, Chuen-Tsai Sun, and Eiji Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall.
3. Simon Haykin, *Neural Networks and Learning Machines*, Pearson.

Reference Books

Reference Books

1. Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, Wiley.
2. David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
3. J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann.
4. S. Rajasekaran and G. A. Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications*, PHI.

Course Outcomes

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

- Explain the fundamental concepts, characteristics, and major techniques of Soft Computing.
- Apply fuzzy sets, fuzzy logic, and fuzzy inference systems to model and solve problems involving uncertainty and approximate reasoning.
- Develop and analyze Artificial Neural Network models for classification, prediction, and pattern recognition problems.
- Apply Genetic Algorithms and swarm intelligence techniques to solve optimization and search problems.
- Design and evaluate hybrid Soft Computing models by integrating fuzzy logic, neural networks, and evolutionary techniques for real-world applications.

Software Engineering

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Software Engineering
Category of Course	Professional Core Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic knowledge of programming, data structures, and databases

Course Objectives

The students will be able to:

- Understand the fundamental principles, processes, and methodologies used in software engineering.
- Apply systematic techniques for software requirements analysis, specification, and modelling.
- Develop software designs using appropriate architectural, structural, and object-oriented design principles.
- Understand software testing, quality assurance, configuration management, and software maintenance practices.
- Apply modern software development methodologies and tools for managing software projects effectively.

Syllabus

UNIT I

Introduction to Software Engineering and Software Processes

Introduction to software engineering; nature and characteristics of software; software applications and software crisis; software engineering as a layered technology; software process concepts; generic process framework; software process models: Waterfall model, Incremental model, Prototyping model, Spiral model, and V-Model; Agile software development; Agile principles and practices; Scrum, Extreme Programming, and DevOps concepts; software process improvement and maturity.

UNIT II

Software Requirements Engineering

Requirements engineering process; stakeholder identification and requirements elicitation; requirements analysis and negotiation; functional and non-functional requirements; Software Requirements Specification (SRS); requirements validation and management; modelling requirements using use-case diagrams, data-flow diagrams, and UML; domain modelling; feasibility study; requirements traceability and change management.

UNIT III

Software Design and Architecture

Fundamentals of software design; design concepts and principles; abstraction, modularity, information hiding, cohesion, and coupling; architectural design and architectural styles; component-level design; data design; interface design; object-oriented design concepts; UML diagrams for design: class diagrams, sequence diagrams, activity diagrams, and state diagrams; design patterns; user interface design principles.

UNIT IV

Software Testing and Quality Assurance

Fundamentals of software testing; testing objectives and principles; levels of testing: unit testing, integration testing, system testing, and acceptance testing; black-box and white-box testing techniques; test case design; equivalence partitioning, boundary value analysis, decision table testing, and path testing; regression testing; test automation concepts; software quality assurance; software quality models; reviews, inspections, and audits; software reliability and metrics.

UNIT V

Software Project Management, Configuration Management, and Maintenance

Software project planning and estimation; software size estimation; cost estimation and COCOMO model; project scheduling and monitoring; risk management; software configuration management; version control and change management; software maintenance and evolution; types of software maintenance; software reengineering and reverse engineering; continuous integration and continuous delivery concepts; software engineering tools and contemporary practices.

Text Books

Text Books

1. Roger S. Pressman and Bruce R. Maxim, *Software Engineering: A Practitioner's Approach*, McGraw-Hill.
2. Ian Sommerville, *Software Engineering*, Pearson.
3. Pankaj Jalote, *An Integrated Approach to Software Engineering*, Springer.

Reference Books

Reference Books

1. Rajib Mall, *Fundamentals of Software Engineering*, PHI Learning.
2. Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli, *Fundamentals of Software Engineering*, Pearson.
3. Stephen R. Schach, *Object-Oriented and Classical Software Engineering*, McGraw-Hill.
4. Bertrand Meyer, *Object-Oriented Software Construction*, Prentice Hall.

Course Outcomes

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

- Explain software engineering principles, software processes, and contemporary software development methodologies.
- Elicit, analyze, specify, validate, and manage software requirements using appropriate models and documentation.
- Apply software design principles, architectural concepts, and UML-based modelling techniques to develop effective software designs.
- Design and evaluate software testing and quality assurance strategies using appropriate testing techniques and quality practices.
- Apply software project management, estimation, configuration management, maintenance, and modern development practices to manage software systems throughout their life cycle.

UI/UX Design Fundamentals

Course Code	Code will be given as per choice in particular semester and paper
Course Name	UI/UX Design Fundamentals
Category of Course	Open Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	No programming knowledge required; basic familiarity with computers and digital applications

Course Objectives

The students will be able to:

- Understand the fundamental concepts of User Interface (UI), User Experience (UX), and human-centred design.
- Identify user needs and apply basic user research and information organization techniques.
- Apply visual and interaction design principles to create simple and effective digital interfaces.
- Develop wireframes, mock-ups, design components, and interactive prototypes using Figma.
- Understand the basic role of HTML, CSS, and JavaScript in converting UI designs into functional web interfaces.

Syllabus

UNIT I

Introduction to UI/UX Design and Human-Centred Design

Introduction to User Interface (UI) and User Experience (UX); Understanding users, user needs, goals, expectations, and contexts of use; introduction to accessibility and inclusive design. Overview of the UI/UX design process from problem identification to prototype evaluation.

Introduction to collaborative design and ideation tools such as Figma, FigJam, and Miro;

UNIT II

User Research, Information Architecture, and Wireframing

Introduction to user research and its role in UX design; basic methods including observation, interviews, questionnaires, and user feedback. Introduction to wireframing and low-fidelity interface design; sketching interfaces and paper prototyping; translating ideas into digital wireframes.

UNIT III

Visual Design and User Interface Design with Figma

Fundamental principles of visual design; layout, alignment, proximity, balance, contrast, consistency, and visual hierarchy; use of whitespace; typography and readability; introduction to colour theory and colour combinations; icons, images, and visual elements. Common user interface components including buttons, input fields, forms, cards, navigation bars, menus, search boxes, and dialogs. Introduction to responsive design concepts and accessibility considerations.

UNIT IV

Introduction to the relationship between UI design and front-end implementation. Basic concepts of HTML for structuring web pages; headings, paragraphs, links, images, lists, forms, and semantic elements. Introduction to CSS for styling interfaces, including colours, fonts, spacing, layout, and responsive design concepts. Introduction to JavaScript for simple user interactions and dynamic behaviour. Understanding how a Figma design is translated into HTML and CSS, with basic awareness of the role of JavaScript in interactive user interfaces. The focus is on introductory understanding rather than advanced programming.

UNIT V

Usability Evaluation and Integrated UI/UX Design Project

Introduction to usability and user experience evaluation; usability principles; Integrated UI/UX design project: identification of a simple real-world user problem; basic user research; preparation of personas and user journeys; definition of user flows and information architecture; creation of wireframes and high-fidelity mock-ups using Figma; development of an interactive prototype; usability evaluation and design refinement. Introduction to the basic workflow for translating selected interface screens into simple HTML and CSS-based web pages, with introductory use of JavaScript concepts for user interaction.

Text Books

Text Books

1. Don Norman, *The Design of Everyday Things*, MIT Press.
2. Steve Krug, *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*, New Riders.
3. Jeff Johnson, *Designing User Interfaces: Strategies for Effective Human-Computer Interaction*, Morgan Kaufmann.

Reference Books

Reference Books

1. Jesse James Garrett, *The Elements of User Experience*, New Riders.
2. Alan Cooper, Robert Reimann, David Cronin, and Christopher Noessel, *About Face: The Essentials of Interaction Design*, Wiley.
3. William Lidwell, Kritina Holden, and Jill Butler, *Universal Principles of Design*, Rockport Publishers.

4. Jenifer Tidwell, Charles Brewer, and Aynne Valencia, *Designing Interfaces*, O'Reilly Media.
5. Jon Duckett, *HTML and CSS: Design and Build Websites*, Wiley.

Course Outcomes

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

- Explain the fundamental concepts of UI, UX, human-centred design, and the stages involved in the UI/UX design process.
- Identify user needs and organize information using basic user research, personas, empathy maps, user journeys, user flows, and information architecture techniques.
- Apply fundamental principles of visual and interface design to create clear, consistent, accessible, and user-friendly digital interfaces using Figma.
- Develop wireframes, high-fidelity mock-ups, reusable interface components, and interactive prototypes using Figma.
- Evaluate and improve digital interfaces using usability principles and user feedback, and demonstrate a basic understanding of how UI designs are realised using HTML, CSS, and introductory JavaScript.

Web and Internet

Course Code	Code will be given as per choice in particular semester and paper
Course Name	Web and Internet
Category of Course	Open Elective Courses
Credits	3L: 0T: 0P C:3
Pre-Requisites	Basic familiarity with computers and Internet usage; no prior programming knowledge required

Course Objectives

The students will be able to:

- Understand the fundamental concepts, architecture, services, and applications of the Internet and the World Wide Web.
- Explain the basic technologies and protocols used for communication and information exchange on the Internet.
- Create and structure simple web pages using HTML and apply CSS for visual presentation and layout.
- Understand basic client-side interactivity using JavaScript and contemporary concepts of responsive web design.
- Develop and publish simple, user-friendly websites by integrating fundamental web technologies and Internet services.

Syllabus

UNIT I

Introduction to Internet and World Wide Web

Introduction to computer networks and the Internet; evolution and growth of the Internet; Internet architecture and basic terminology; Internet Service Providers (ISPs); IP addresses and domain names; Uniform Resource Locator (URL); Domain Name System (DNS); client-server architecture; introduction to the World Wide Web; web browsers and web servers; search engines and web search; Internet services including e-mail, file transfer, cloud-based services, and online communication. Introduction to Internet protocols and the TCP/IP model; overview of HTTP and HTTPS.

UNIT II

Web Fundamentals and HTML

Introduction to web technologies and web development; static and dynamic websites; structure of a web page; introduction to HTML and HTML document structure; headings, paragraphs, formatting elements, links, images, lists, tables, and multimedia elements; forms and basic form controls; semantic HTML elements; introduction to accessibility in web pages. Basic concepts

of web hosting and domain registration. Hands-on development of simple multi-page websites using HTML.

UNIT III

Web Design and Cascading Style Sheets

Principles of effective web design; usability, readability, consistency, visual hierarchy, colour, typography, and page layout. Introduction to Cascading Style Sheets (CSS); selectors, properties, values, and cascading; colours, backgrounds, fonts, text formatting, borders, margins, padding, and the box model. Styling links, lists, tables, forms, and images; introduction to CSS layout techniques using Flexbox and Grid; responsive web design concepts; media queries and designing web pages for different screen sizes. Hands-on development and styling of responsive web pages.

UNIT IV

Introduction to JavaScript and Interactive Web Pages

Introduction to client-side scripting and JavaScript; basic syntax, variables, data types, operators, expressions, and control statements; functions and events; introduction to the Document Object Model (DOM); selecting and modifying web page elements; event handling and basic form validation; simple interactive user interfaces; introduction to browser developer tools and debugging. Integration of HTML, CSS, and JavaScript for developing simple interactive web applications. Overview of web security awareness, including safe browsing, secure passwords, and basic concepts of client-side security.

UNIT V

Web Applications, Publishing, and Contemporary Web Technologies

Introduction to web applications and their basic architecture; front-end and back-end concepts; introduction to databases in web applications; overview of web APIs and data exchange concepts; introduction to content management systems and website builders. Web hosting, domain names, and website deployment; basic website testing and maintenance; performance, accessibility, and usability considerations. Introduction to contemporary web technologies including Progressive Web Applications, cloud-based web services, and mobile-friendly web design. Integrated mini-project: planning, designing, developing, testing, and publishing a simple responsive and interactive website.

Text Books

Text Books

1. Jon Duckett, *HTML and CSS: Design and Build Websites*, Wiley.
2. Jon Duckett, *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley.
3. Jeffrey C. Jackson, *Web Technologies: A Computer Science Perspective*, Pearson.

Reference Books

Reference Books

1. Thomas A. Powell, *HTML & CSS: The Complete Reference*, McGraw-Hill.

2. Robin Nixon, *Learning PHP, MySQL & JavaScript*, O'Reilly Media.
3. Paul Deitel and Harvey Deitel, *Internet and World Wide Web: How to Program*, Pearson.
4. Jennifer Niederst Robbins, *Learning Web Design*, O'Reilly Media.

Course Outcomes

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

- Explain the architecture, services, terminology, and fundamental technologies of the Internet and the World Wide Web.
- Create structured and accessible web pages using HTML and organize web content effectively.
- Apply CSS principles and layout techniques to design visually appealing, responsive, and user-friendly websites.
- Develop simple interactive web pages using basic JavaScript, DOM manipulation, events, and form validation.
- Design, develop, test, and publish a simple responsive web application or website by integrating HTML, CSS, JavaScript, and fundamental Internet technologies.