VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CURRICULUM AND SYLLABI

(2026-2027)

M. Tech Computer Science and Engineering (Data Science)-5 Year Integrated

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech Computer Science and Engineering (Data Science)-5Year Integrated

VISION STATEMENT OF VELLORE INSTITUTE OF TECHNOLOGY

Transforming life through excellence in education and research.

MISSION STATEMENT OF VELLORE INSTITUTE OFTECHNOLOGY

World class Education: Excellence in education, grounded in ethics and critical thinking, for improvement of life.

Cutting edge Research: An innovation ecosystem to extend knowledge and solve critical problems.

Impactful People: Happy, accountable, caring and effective workforce and students.

Rewarding Co-creations: Active collaboration with national & international industries & universities for productivity and economic development.

Service to Society: Service to the region and world through knowledge and compassion.

VISION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

To be a world-renowned centre of education, research and service in computing and allied domains.

MISSION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

- To offer computing education programs with the goal that the students become technically competent and develop lifelong learning skills.
- To undertake path-breaking research that creates new computing technologies and solutions for industry and society at large.
- To foster vibrant outreach programs for industry, research organizations, academia and society.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech Computer Science and Engineering (Data Science)-5Year Integrated

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. Graduates will be engineering professionals who will engage in technology development and deployment with social awareness and responsibility.

2. Graduates will function as successful practice engineer / researcher / teacher / entrepreneur in the chosen domain of study.

3. Graduates will have holistic approach addressing technological, societal, economic and sustainability dimensions of problems and contribute to economic growth of the country.

PROGRAMME OUTCOMES (POs)

PO_01: Having an ability to apply mathematics and science in engineering applications.

PO_03: Having an ability to design a component or a product applying all the relevant standards and with realistic constraints, including public health, safety, culture, society and environment

PO_04: Having an ability to design and conduct experiments, as well as to analyze and interpret data, and synthesis of information

PO_05: Having an ability to use techniques, skills, resources and modern engineering and IT tools necessary for engineering practice

PO_06: Having problem solving ability- to assess social issues (societal, health, safety, legal and cultural) and engineering problems

PO_07: Having adaptive thinking and adaptability in relation to environmental context and sustainable development

PO_08: Having a clear understanding of professional and ethical responsibility

PO_11: Having good cognitive load management skills related to project management and finance

ADDITIONAL PROGRAMME OUTCOMES (APOs)

APO_02: Having Sense-Making Skills of creating unique insights in what is being seen or observed (Higher level thinking skills which cannot be codified)

APO_03: Having design thinking capability

APO_04: Having computational thinking (Ability to translate vast data in to abstract concepts and to understand database reasoning

APO_07: Having critical thinking and innovative skills

APO_08: Having a good digital footprint

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

M. Tech Computer Science and Engineering (Data Science)-5Year Integrated

PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. Apply mathematical models, engineering principles, and efficient data structures and algorithms to develop solutions for real-world problems.

2. Design and implement scalable data storage systems, and employ statistical and computational methods for data analysis and knowledge representation.

3. Utilize data management, analytics, and visualization tools to transform raw data into actionable knowledge for real-world decision-making across domains.

CURRICULUM AND SYLLABUS
2026-2027

Category Credit Detail			
Sl.No.	**Description**	**Credits**	**Maximum Credit**
1	UCC - universityCore Course	72	72
2	PC- ProfessionalCoreCourse	80	80
3	PEC - PROFESSIONAL Elective Course	28	28
4	OE - Open Elective	20	20
Total Credits		200	

University Core Course

sl.no	Course Code	Course Title	Course Type	Version	L	T	P	J	Credits
1	IACHY102	Engineering Chemistry	Theory Only	1.0	3	0	2	0	4
2	IAPHY101	Engineering Physics	Theory Only	1.0	3	0	2	0	4
3	IAMAT101	Multivariable Calculus and Differential Equations	Theory Only	1.0	3	0	2	0	4
4	IAMAT201	Applied Linear Algebra	Theory Only	1.0	3	1	0	0	4
5	IAMAT202	Discrete Mathematics and Graph Theory	Theory Only	1.0	3	1	0	0	4
6	IAMAT203	Probability and Statistics	Theory Only	1.0	3	0	2	0	4
7	IAEEE101	Basic Electrical and Electronics Engineering	Theory Only	1.0	3	0	2	0	4
8	IACSE101	Problem Solving Using Python	Theory Only	1.0	0	0	4	0	2
9	IACSE102	Problem Solving Using Java	Theory Only	1.0	0	0	4	0	2
10	IACSE201	Structured and Object-Oriented Programming	Theory Only	1.0	2	0	4	0	4
11	IASTS101	Qualitative and Quantitative Skills Practice I	Theory Only	1.0	3	0	0	0	1
12	IASTS102	Qualitative and Quantitative Skills Practice II	Theory Only	1.0	3	0	0	0	1
13	IASTS201	Qualitative and Quantitative Skills Practice III	Theory Only	1.0	3	0	0	0	1
14	IASTS202	Qualitative and Quantitative Skills Practice IV	Theory Only	1.0	3	0	0	0	1
15	IAFLC100	Foreign Language	Theory Only	1.0	1	0	2	0	2
16	IAHUM101	India Studies	Theory Only	1.0	1	0	0	0	1
17	IACHY101	Environmental Sciences	Theory Only	1.0	2	0	0	0	2
18	IAHSM100	Humanities, Social Science and Management	Theory Only	1.0	3	0	0	0	3
19	IAENG101	Technical English Communication	Theory Only	1.0	3	0	2	0	4
20	IACSI399	Summer Industrial Internship	Project	1.0	0	0	4	0	2
21	IACSI498	Project – I	Project	1.0	0	0	16	0	8
22	IACSI499	Project – II / Capstone	Project	1.0	0	0	20	0	10
23	IAENG102	Effective English Communication (NCC)	Theory Only	1.0	0	0	4	0	2
24	IAEXC100	Extracurricular Activities (NCCM)	Project	1.0	0	0	4	0	2

Professional Core Courses									
sl.no	**Course Code**	**Course Title**	**Course Type**	**Version**	**L**	**T**	**P**	**J**	**Credits**
1	IAMAT204	Complex Variables and Integral Transforms	Theory Only	1.0	3	1	0	0	4.0
2	IACSI101	Data Structures and Algorithm Analysis	Theory Only	1.0	3	0	2	0	4.0
3	IACSI201	Design Structure and Algorithms	Theory Only	1.0	3	0	2	0	4.0
4	IACSI202	Digital Logic and Computer Design	Theory Only	1.0	3	0	2	0	4.0
5	IACSI203	Advanced Computer Architecture and Organization	Theory Only	1.0	3	1	0	0	4.0
6	IACSI204	Database Systems Design	Theory Only	1.0	3	0	2	0	4.0
7	IACSI205	Operating Systems Fundamentals	Theory Only	1.0	3	0	2	0	4.0
8	IACSI206	Internet and Web Programming	Theory Only	1.0	2	0	4	0	4.0
9	IACSI301	Microprocessors and Microcontrollers	Theory Only	1.0	3	0	2	0	4.0
10	IACSI302	Data Communication and Computer Networks	Theory Only	1.0	3	0	2	0	4.0
11	IACSI303	IoT and Embedded Systems	Theory Only	1.0	3	0	2	0	4.0
12	IACSI304	Cryptography and Cyber Security	Theory Only	1.0	3	0	2	0	4.0
13	IACSI305	Artificial Intelligence and Machine Learning	Theory Only	1.0	3	0	2	0	4.0
14	IACSI306	Software Engineering Principles and Practices	Theory Only	1.0	3	0	2	0	4.0
15	IACSI308	Formal Language and Automata Theory	Theory Only	1.0	3	1	0	0	4.0
16	IACSI401	Cloud Computing Technologies	Theory Only	1.0	3	0	2	0	4.0
17	IAMDI401	Programming for Data science	Theory Only	1.0	3	0	2	0	4.0
18	IAMDI402	Data Visualization	Theory Only	1.0	3	0	2	0	4.0
19	IAMDI403	Data Mining	Theory Only	1.0	3	0	2	0	4.0
20	IAMDI404	Soft Computing Techniques	Theory Only	1.0	3	1	0	0	4.0

Professional Elective Courses									
sl.no	**Course Code**	**Course Title**	**Course Type**	**Version**	**L**	**T**	**P**	**J**	**Credits**
1	IAMDI405	Machine Learning for Data Science	Theory Only	1.0	3	0	2	0	4.0
2	IAMDI406	Distributed Systems	Theory Only	1.0	3	1	0	0	4.0
3	IAMDI407	Exploratory Data Analysis	Theory Only	1.0	3	0	2	0	4.0
4	IAMDI408	Statistical Inference	Theory Only	1.0	3	0	2	0	4.0
5	IAMDI409	Predictive Analytics	Theory Only	1.0	3	1	0	0	4.0
6	IAMDI410	Healthcare Data Analytics	Theory Only	1.0	3	1	0	0	4.0
7	IAMDI411	Financial Data Analytics	Theory Only	1.0	3	0	2	0	4.0
8	IAMDI412	Advanced Data Visualization Techniques	Theory Only	1.0	3	0	2	0	4.0
9	IAMDI413	Advanced Data Compression Techniques	Theory Only	1.0	3	0	2	0	4.0
10	IAMDI414	Computer Vision	Theory Only	1.0	3	0	2	0	4.0
11	IACSI404	Deep Learning	Theory Only	1.0	3	0	2	0	4.0
12	IACSI405	MLOPs	Theory Only	1.0	3	0	2	0	4.0
13	IACSI406	GenAI and Large Language Models	Theory Only	1.0	3	0	2	0	4.0
14	IACSI407	Agentic AI	Theory Only	1.0	3	0	2	0	4.0

Course Code	Course Title	L	T	P	C
IACHY102	Engineering Chemistry	3	0	2	4
Pre-requisite	no	**Syllabus Version**			
		1			

Course Objectives

To enable students to learn fundamental concepts in different branches of chemistry
To be well-versed in analytical and computational skills in chemistry
To teach practical skills in chemistry and apply knowledge for societal applications

Course Outcomes

Apply the concepts of thermodynamics and kinetics (BL3)
Use chemical concepts for the advancement of materials in electrical, magnetic, and energy domains (BL3)
Demonstrate UV Visible spectroscopic and X- ray diffraction techniques (BL3)
Discuss molecular orbitals and its energies through computational chemistry (BL2)
Apply instrumental, wet-chemical, and computational methods to analyze materials, molecules, and ions (BL3)

Module:1	**Chemical thermodynamics and kinetics**	**9 hours**

Introduction-Terminologies- Zeroth law- First law of thermodynamics - Enthalpy and heat capacity - Applications of first law of thermodynamics - Second law of thermodynamics - Carnot cycle- Third law of thermodynamics - Gibbs Helmholtz equation. Chemical Kinetics - Rate equation - Order of the reaction - Integral rate equations - Half life time of reactions- Molecularity - Collision theory of reaction rate - Effect of temperature and catalysts on reaction rate -Numericals.

Module:2	**Energy devices**	**8 hours**

Electrochemical and electrolytic cells -electrode materials with examples (semi-conductors), classifications of batteries - chemistry of Li - ion secondary batteries, Fuel cells: H_2-O_2 and solid oxide fuel cell (SOFC); Solar cells - photovoltaic cell (silicon based), photoelectrochemical cells and dye-sensitized cells.

Module:3	**Functional materials**	**10 hours**

Electrical materials: Ohm's law, resistivity, classification - metals, semiconductors (intrinsic & extrinsic-Si), insulators, superconductors. Magnetic materials: para, dia, ferro & anti-ferro magnetism, applications in storage devices. Liquid cooling **-** types of coolants, thermal conductivity, and applications. Nanomaterials: introduction, top-down and bottom-up approaches for synthesis - bulk *vs* nano (gold), quantum dots.		
Module:4	**Spectroscopic and diffraction techniques**	**6 hours**
Fundamental concepts in spectroscopic and diffraction techniques - Principle, instrumentation, and applications of UV-Visible and powder XRD (determination of crystallite size, numericals)		
Module:5	**Computational Chemistry Basics to Applications**	**10 hours**
Introduction-Quantum mechanical calculations. Molecular orbital theory: atomic orbitals, molecular orbitals. Orbital interactions- chemical bonding. Potential energy curve/surface, electrostatic potentials, geometry optimization, frequency, reaction mechanism, application of AI to screen semiconductors		
Module:6	**Contemporary Topics**	**2 hours**
Contemporary Topics by Industry Expert		
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Theodore E. Brown, H Eugene, LeMay Bruce E. Bursten, Catherine Murphy, Patrick Woodward, Matthew E. Stoltzfus, " **Chemistry: The Central Science**", Pearson Publishers, 14th Edition, **2017** Frank Jensen, "**Introduction to computational chemistry**", Wiley , 3rd Edition, **2017**		
Reference Books		
A.R West, "**Solid State Chemistry and its Applications**", Wiley, 2nd Edition, **2014** Lawrence S. Brown and Thomas Holme, Chemistry for engineering students, 2018, 4th edition. "**Chemistry for engineering students**", Brooks Cole, 4th Edition, **2018**		
Indicative Experiments		
1. Thermodynamic functions from EMF measurements: Zn - Cu system. Evaluation of properties enthalpy, entropy, and free energy.		2 hours
2. Determination of reaction rate, order and molecularity - ester hydrolysis (ethylacetate)		2 hours

3. Colorimetric estimation of Ni^{2+} using conventional and smartphone digital-imaging method.	2 hours
4. Analysis of iron in carbon steel by potentiometry	2 hours
5. Preparation of ZnO semiconductor and its characterization.	2 hours
6. Estimation of sulfate ion in drinking water by the conductivity method	2 hours
7. Build atoms and molecules, visualization of atomic orbitals and hybridized orbitals. Calculating the orbital contributions.	2 hours
8. Conformational analysis of cyclohexane and ethane molecules and plotting the potential energy profile.	2 hours
9. Application of AI to screen semiconductors, materials, drugs - case study	2 hours
10. Size-dependent colour variation of Cu_2O nanoparticles by spectrophotometer	2 hours
11. Laboratory synthesis of hydroxyl apatite and its use for the removal of heavy metals	2 hours
12. Challenging experiment related to evaluation of the state of charge of batteries (SOC)	2 hours
13. Construction of DSSC and its working	2 hours
14. Preparation of Ag metal nanoparticles and their characterization	2 hours
15. Colorimetric estimation of Fe^{2+} using conventional and smartphone digital-imaging method	2 hours
Total Laboratory Hours:	**30 hours**
Text Book(s)	
Vogel, "**Vogel's Quantitative Chemical analysis**", Pearson Education, 6th Edition, **2008**	
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
	Engineering Physics	**3**	**0**	**2**	**4**
Pre-requisite	**-**	**Syllabus Version**			
		1.0			

Course Objectives

1. To explain the dual nature of radiation and matter.
2. To apply Schrödinger's equation to solve finite and infinite potential problems.
3. To explain various components of optical fiber communication system.

Course Outcomes

At the end of this course, the student will be able to

1. **Explain** wave-particle duality by applying basic quantum principles.
2. **Apply** quantum mechanical ideas to microscopic systems.
3. **Examine** the characteristics and working of lasers by applying the Einstein's theory of stimulated emission.
4. **Describe** the primary components of a typical optical fiber communications system and their working principles.
5. **Demonstrate** the quantum phenomena and optical fiber communication though experiments and simulations

Module: 1	**Elements of Quantum Mechanics**	**10 hours**

Idea of Quantization - Compton effect – de Broglie hypothesis - Davisson-Germer experiment - Wave function and probability interpretation - Heisenberg uncertainty principle - Schrödinger wave equation - time dependent and time independent forms.

Module: 2	**Applications of Quantum Mechanics**	**7 hours**

Eigenvalues and eigenfunction of particle confined in one dimensional box - Basics of nanophysics - Quantum confinement and nanostructures - Tunnel effect - scanning tunnelling microscope

Module: 3	**Laser**	**10 hours**

Laser characteristics - spatial and temporal coherence - Einstein coefficients and their significance - Population inversion - two, three and four level systems - Pumping schemes - threshold gain coefficient - Components of a laser - He-Ne, Nd:YAG and CO_2 lasers and their engineering applications.

Module: 4	**Fiber Optics**	**8 hours**

Introduction to optical fiber communication system - light propagation through fibers - Acceptance angle - Numerical aperture - V-parameter - Types of fibers – Attenuation - Dispersion-intermodal and intramodal.

Module: 5	**Optoelectronic Devices**	**8 hours**

Introduction to semiconductors - direct and indirect bandgap – Sources: LED and laser diode, Photodetectors: PN and PIN.

Module: 6	**Contemporary Topics**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Books

1	H. D. Young and R. A. Freedman, University Physics with Modern Physics, 2020, 15th Edition, Pearson, USA.
2	D. K. Mynbaev and Lowell L. Scheiner, Fiber Optic Communication Technology, 2011, 1st Edition, Pearson, USA

Reference Books

1.	Robert Eisberg and Robert Resnick, Quantum Physics, 2004, 2nd Edition. Wiley, India

Mode of Evaluation: Quiz, Assignment, CAT and FAT

List of Challenging and Mini Projects - CHAMP (Indicative)

1.	Modeling time dependent behaviour of quantum wave packets.
2.	Model tunneling through barriers of different profiles and study dependence on particle energy.
3.	Exploring the wave nature of electrons through diffraction.
4.	Determine the fundamental quantum constant through the analysis of semiconductor junction characteristics.
5.	Elemental analysis through photoelectron spectroscopy.
6.	Unveiling stellar composition through quantum simulations of hydrogen spectral lines.
7.	Designing a solar cell based renewable energy device circuits for electric power generation.
8.	Analyze photodiode I–V characteristics under different light intensities to extract device parameters.
9.	Visualize the uncertainty principle by simulating position-momentum evolution of quantum wave packets.
10.	Crystallite size estimation from XRD pattern.
11.	Experimental determination of electrical band gap in semiconductors: Insights into band theory and semiconductor behaviour.
12.	Verification of Maxwell's equations using experimental data.
13.	Analysing the dispersive power of a prism.
14.	Designing a monochromator using diffraction technique.
15.	Simulate Compton scattering and demonstrate how the change in photon wavelength depends on the scattering angle.

Total Laboratory Hours	**30 hours**

Text Books

1.	H.D. Young and R. A. Freedman, University Physics with Modern Physics, 2020, 15th Edition, Pearson, USA

Reference Books

1.	R. A. Serway, J. W. Jewett, Jr. Physics for Scientists and Engineers with Modern Physics, 2019, 10th Edition, Cengage Learning, USA

Mode of assessment: Continuous assessment, FAT, Oral examination

Recommended by Board of Studies	DD-MM-YYYY		
Approved by Academic Council	No. xx	Date	DD-MM-YYYY

Course Code	Multivariable Calculus and Differential Equations	L	T	P	C
IAMAT101		3	0	2	4
Pre-requisite		Syllabus Version			

Course Objectives

1. To provide essential foundational knowledge in engineering mathematics, supporting engineers and scientists in mastering advanced mathematical concepts relevant to their field
2. To enhance problem-solving skills for engineers and scientists by mastering vector calculus principles, including gradient, divergence, curl, and fundamental theorems, for real-world engineering applications.
3. To develop expertise in solving differential equations through various techniques, enabling engineers and scientists to address complex challenges in system modeling and simulations.

Course Outcomes

At the end of the course, the student is able to:

1. evaluate partial derivatives and solve optimization problems involving several variables with or without constraints.
2. solve multiple integrals in cartesian, polar, cylindrical and spherical coordinates.
3. understand gradient, directional derivatives, divergence, curl, Green's, Stokes and Gauss Divergence theorems.
4. utilize different solution methods to obtain the solution for second-order ordinary differential equations.
5. demonstrate proficiency in solving linear and nonlinear first-order partial differential equations.

Module:1	Functions of Several Variables	8 hours

Functions of two variables- Partial derivatives –total differential -Jacobian and its properties. Taylor's expansion for two variables–maxima and minima–constrained maxima and minima-Lagrange's multiplier method.

Module:2	Multiple Integrals	9 hours

Evaluation of double integrals–change of order of integration–change of variables between Cartesian and polar co-ordinates - Evaluation of triple integrals - change of variables between Cartesian and cylindrical and spherical co-ordinates.

Module:3	Vector Differentiation and Integration	10 hours

Scalar and vector valued functions – gradient, directional derivative, divergence, and curl – scalar and vector potentials. Line, surface and volume integrals -Statement of Green's, Stoke's and Gauss divergence theorems -Verification and evaluation of vector integrals using them.

Module:4	Ordinary Differential Equations	8 hours

Linear second-order ordinary differential equations with constant coefficients – Solutions of ODE by Method of undetermined coefficients – Method of variation of parameters – Solutions of Cauchy-Euler and Cauchy-Legendre differential equations – Applications of linear second-order differential equations.

Module:5	Partial Differential Equations	8 hours

Formation of PDE- solution of first order partial differential equation of the forms: $F(p,q)=0$, $F(z,p,q)=0$, $F(x,p)=G(y,q)$ and Clairaut's form - Lagrange's equation: $Pp+Qq = R$. Solution of a linear first order partial differential equation by separation of variables.

Module:6	Contemporary Topics	2 hours

	Total Lecture hours:			**45 hours**
Text Books				
1	Erwin Kreyszig, Advanced Engineering Mathematics, 10th Edition, Wiley India, 2015.			
2	B.S. Grewal, Higher Engineering Mathematics, 44th Edition, Khanna Publishers, 2020			
Reference Books				
1.	George B. Thomas, D. Weir and J. Hass, Thomas' Calculus, 15th edition, Pearson, 2023.			
2.	Dennis G. Zill, Advanced Engineering Mathematics, 6th Edition, Jones & Bartlett Publishers, 2018.			
Mode of Evaluation: Quizzes, Assignments, CATs and FAT				
List of Experiments (Indicative)				
1.	Introduction to Calculus through MATLAB – Symbolic Computations			
2.	Plotting and visualizing curves and surfaces			
3.	Evaluating maxima and minima of functions of several variables			
4.	Applying Lagrange multiplier optimization method			
5.	Evaluation of volume under a surface by double Integral			
6.	Evaluating triple integrals			
7	Demonstration of gradient, divergence and curl			
8	Evaluating line integrals in vectors			
9	Solution of Linear differential equations			
10	Solution of Partial differential equations			
			Total Lab hours:	**30 hours**
Text Book				
1	Ronald L. Lipsman , Jonathan M. Rosenberg, Multivariable Calculus with MATLAB with applications to Geometry and Physics, Springe, 2017.			
Reference Book				
1	Dean G. Duffy, Advanced Engineering Mathematics with MATLAB, CRC Press, 2022.			
Mode of Evaluation: Assignments and FAT				
Recommended by Board of Studies		14-May-2025		
Approved by Academic Council		**No.**	**Date**	

Course Code	Course Title	L	T	P	C
IAMAT201	Applied Linear Algebra	3	1	0	4
Pre-requisite	IAMAT101	Syllabus Version			
		1.0			

Course Objectives

1. To develop expertise in matrix operations, eigenvalue analysis, and system-solving techniques for efficient modeling and problem-solving in engineering applications.
2. To apply differential equation-solving methods using matrix approaches to analyze dynamic systems such as mechanical vibrations and electrical circuits.
3. To utilize vector spaces, linear transformations, and inner product spaces to optimize computational processes and enhance engineering analysis.

Course Outcomes

1. Solve the linear system of equations.
2. Utilize matrix methods to solve system of ordinary differential equations.
3. Demonstrate the concepts of vector spaces and their applications.
4. Understand the concepts of linear transformations.
5. Apply vector normalization techniques to generate orthonormal basis.

Module:1	Matrices and Systems of Linear Equations	8 hours

Eigen values and Eigen vectors - properties of Eigen values and Eigen vectors- Cayley Hamilton theorem - Gaussian elimination and Gauss Jordan methods - Elementary matrices- permutation matrix - inverse matrices - System of linear equations - LU factorization.

Module:2	Matrix Methods to Solve ODE	8 hours

Solving a nonhomogeneous first order and homogeneous second-order system of differential equations, Reduction of an n^{th} order differential equation to a first-order system**.** Applications – Mass-spring systems, Series LCR circuits.

Module:3	Vector Spaces	9 hours

The Euclidean space and vector space- subspace –linear combination-span-linearly dependent-independent- bases - dimensions-finite dimensional vector space. Row and column spaces -Rank and nullity – Bases for subspace – invertibility- Application in interpolation.

Module:4	Linear Transformations	9 hours

Linear transformations – Basic properties- invertible linear transformation - matrices of linear transformations - vector space of linear transformations – change of bases – similarity transformation.

Module:5	Inner Product Spaces	9 hours

Dot products and inner products – the lengths and angles of vectors – matrix representations of inner products- Gram-Schmidt orthogonalization - Orthogonal transformation and nature of quadratic form, QR factorization- Projection - orthogonal projections – relations of fundamental subspaces –Least Square solutions in Computer Codes.	
Total Lecture Hours:	**43 hours**
Tutorial Hours:	**15 hours**
Text Book(s)	
1. David C. Lay, "**Linear Algebra and Its Applications**", Pearson, 5th Edition, **2019** 2. Bernard Kolman, David R. Hill, "**Elementary Linear Algebra with Applications**", Pearson Education, 9th Edition, **2023**	
Reference Books	
1. Stephen Andrilli, David Hecker, "**Elementary Linear Algebra**", Academic Press, 5th Edition, **2016** 2. Gilbert Strang, "**Introduction to Linear Algebra**", Wellesley Cambridge Press, 6th Edition, **2023**	
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test	
Recommended by Board of Studies :	14-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
IAEEE101	**Basic Electrical and Electronics Engineering**	3	0	2	4
Pre-requisite	NIL	Syllabus version			
		v.2			

Course Objectives

1. Familiarize with various laws and theorems to solve electric and electronic circuits
2. Provide an overview on working principle of machines
3. Understand the concepts of semiconductor devices, op-amps and digital circuits

Course Outcomes

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

1. Apply various circuit laws and theorems for electric circuit analysis.
2. Determine the parameters of magnetic circuits.
3. Explain the construction, working principle and applications of various types of electrical machines.
4. Design basic combinational circuits in digital systems.
5. Illustrate the characteristics and applications of semiconductor devices.

Module:1	DC Circuits	9 hours

Basic circuit elements and sources; Ohms law; Kirchhoff's laws; Series and Parallel connection of circuit elements; Mesh current analysis; Node voltage analysis; Source Transformation, Thevenin's theorem, Maximum power transfer theorem and Superposition theorem.

Module:2	AC and Magnetic Circuits	9 hours

Alternating voltages and currents, RMS, average, maximum values, Form Factor, Peak Factor, Instantaneous and Average Power, Single Phase RL, RC, RLC series circuits, Power in AC circuits, Power Factor,

Magnetic circuits: Toroidal core: Flux density, Permeability, Flux linkage; Magnetic circuit with airgap; Reluctance in series and parallel circuits; Self and mutual inductance; Dot Convention

Module:3	Electrical Machines	9 hours

Construction, working principle and applications of DC Generator and Motor, Single phase Transformers, Three phase Induction motors, Three phase Alternators, single phase induction motors, stepper motor, and BLDC motor.

Module:4	Digital Systems	8 hours

Number base conversion; Boolean algebra: simplification of Boolean functions using K-maps-SOP & POS; Don't Care Conditions, Logic gates; Design of basic combinational circuits: adders, multiplexers, de-multiplexers.

Module:5	Electronic Devices and Circuits	8hours

Characteristics: PN junction diode, Zener diode, BJT, MOSFET; Applications: single phase full wave Rectifier, Voltage regulator, Amplifiers.

Module:6	Contemporary Topics	2 hours

	Total Lecture hours:	45 hours

Text Books

1	Allan R. Hambley, "Electrical Engineering -Principles & Applications", 2019, 6th Edition, Pearson Education
2	V. D. Toro, Electrical Engineering Fundamentals, 2nd edition. PHI, 2014

Reference Books

1	R. L. Boylestad and L. Nashelsky, Electronic Devices and Circuit Theory, 11th edition. Pearson Publications

2	DP Kothari & Nagrath, "Basic Electric Engineering", 2019, Tata McGraw Hill
	Suggested list of Lab Experiments (Electrical and Electronics) 1. Verification of mesh and nodal analysis for a given electrical circuit 2. Verification of Thevenin's Theorem for a given electrical circuit 3. To conduct an experiment to verify Maximum power transfer theorem 4. Conduct an experiment for Sinusoidal steady state analysis of series RLC circuit 5. Verification of resonance in series RLC circuit 6. Conduct an experiment for Measurement of Power and Power Factor in series RL circuit. 7. Measurement of Earth Resistance Using Megger 8. Conduct an experiment on Zener Voltage Regulator 9. Conduct an experiment on Single phase full wave diode Rectifier with R-Load 10. Conduct an experiment on Light dimmer circuit using Darlington pair of BJTs 11. Determine, self-inductance, mutual inductance and coefficient of coupling of a coupled magnetic circuit. 12. Design of binary adder circuits.

Mode of Evaluation: CAT, Assignment, Quiz, Seminar, Project, FAT			
Recommended by Board of Studies	23-05-2025		
Approved by Academic Council	No.	Date	

Course Code	Course Title	L	T	P	C
IACSE101	Problem Solving Using Python	0	0	4	2
Pre-requisite	Nil	Syllabus Version			
		1.0			

Course Objectives

To build a solid foundation in Python programming by learning essential coding constructs and logic development.
To enhance the skill to write efficient, modular python programs for solving computational problems.
To impart knowledge on modular programs using functions, manipulate data using key Python structures, and ensure reliability through file and error handling

Course Outcomes

Apply fundamental programming concepts such as variables, data types, control flow (conditional statements and loops) to solve basic computational problems using Python
Design and develop modular, reusable, and efficient code using functions and Python's built-in data structures like lists, tuples, dictionaries, and sets to manipulate and process data.
Build robust and functional Python applications by implementing file operations, exception handling, and by leveraging Python libraries and packages for advanced tasks such as data analysis, automation, and mathematical computations.

Indicative Experiments

1. **Module:1 Introduction to Problem solving and Python Programming Fundamentals** Problem Solving: Definition and Steps, Problem solving Approaches, Algorithms, Flowcharts and Pseudo code, Python setup, Syntax, and Simple Input and Output Statements, Modes of Execution, Indentation, Comments, Variables, Data types, Operators and their precedence, Expression. **List of Experiments (Indicative)** **Designing an algorithm for a vending Machine (M1-CO1)** The machine sells items priced at $1.50. It accepts only $1, $2, and $5 bills. The machine should: • Accept the inserted bill. • Check if the bill is valid. • If the bill is less than $1.50, display "Insufficient amount. Please insert more." • If the bill is equal to $1.50, display "Thank you. No change." • If the bill is more than $1.50, calculate and return the change. • If the bill is invalid (not $1, $2, or $5), display "Invalid bill. Please insert a valid amount." **Task:**	4 hours

1. Draw a flowchart and write an algorithm that accepts the inserted bill and processes the transaction accordingly. 2. Include decision-making steps for: o Validity check of the bill o Comparison of inserted amount with item price o Change calculation (if needed)	
2. **Module:1 Introduction to Problem solving and Python Programming Fundamentals** Problem Solving: Definition and Steps, Problem solving Approaches, Algorithms, Flowcharts and Pseudo code, Python setup, Syntax, and Simple Input and Output Statements, Modes of Execution, Indentation, Comments, Variables, Data types, Operators and their precedence, Expression. **List of Experiments (Indicative)** **Calculation of amount in a Piggy bank (M1-CO1)** A Kid has a hobby of collecting coins and rupee notes (Assume the denominations to be one rupee, two rupees, five rupees and ten rupees) in a piggy bank. Once in a year, he calculates and writes down the total amount collected by him in a sheet and use the amount to get a comic book. (Assume that the collected amount is always greater than the price of comic book.) Write a program to automate the process of counting the total amount collected. Also, display the extra amount to be returned to piggy bank. Print 'Invalid input' for input outside the boundary conditions mentioned in the problem. **Input Forma**t: The first input value read is the number of one rupee coins in the piggy bank. The second input value read is the number of two rupees coins in the piggy bank. The third input value read is the number of five rupees coins in the piggy bank. The fourth input value read is the number of ten rupees coins in the piggy bank. The fifth input value read is the money required to purchase the comic book. **Output Format:** The total amount of money saved by the kid can be calculated using the formula, number of one rupee coins + (number of two rupees coins * 2) + (number of five rupees coins * 5) + (number of ten rupees coins* 10) The amount to be kept again in piggy bank can be calculated using by subtracting the money required for purchasing the comic book from the total amount saved. **Boundary Conditions: >=0** Example Input/Output 1: Input: 5 4 3 5 75 Output: 78 03 **Explanation:** In the input, there are five one rupee, 4 two rupees, 3 five rupees and 5 ten rupees coins.	4 hours

Hence the total amount is calculated as (5*1) + (4*2) +(3*5) + (5*10) Money to be returned to piggy bank = 78 – 75 = 03 Example Input/Output 2: Input: 10 0 4 10 120 Output: 130 10 Explanation: (10*1) + (0*2) +(4*5) + (10*10) 130-120 = 10 **Test cases** Input (Each input is separated by a comma) 10 -3 5 6, 20 30 5 3, 34 35 12 5 Output (Each input is separated by $END) Invalid input$END 135$END 214$END	
3. **Module:2 Program Flow Mechanisms** Conditional Statements: if, if-else, nested if, multi-way if-elif statements, Looping: while loop, for loop, else clauses in loops, nested loops, Loop Control Statements: break, continue and pass statements. **List of Experiments (Indicative)** **Employee appraisal process (M2-CO1)** Sun Microsystems Limited has taken a decision to appraise the salary of the employee as follows: If the total work experience is more than 10 years then 10% hike to the existing salary. If the total work experience is more than 5 then 5% hike to the existing salary. Otherwise 3% hike to the existing salary. Write a program to display the revised salary of the employee. **Input format:** Read the employee current salary Read the experience of the employee **Output format:** Revised salary of the employee Boundary Condition: Salary and experience should be >= 0 **Input :** 25000 4 $END 30000 12 **Output :** 25750 $END 33000	4 hours
4. **Module:2 Program Flow Mechanisms** Conditional Statements: if, if-else, nested if, multi-way if-elif statements, Lo pass statements. **List of Experiments (Indicative)**	oping: while lo 4 hours

op, for loop, else cla

Smallest Integer (M2-CO1) Given a number 'p', develop a logic and write a Python program to determine the smallest integer 'n' in the sum of greater than 'p'. Print the value of 'n' and the sum 's'. For example, when p=3, the sum of the reciprocals continue sum‘s’ is 3.02 **Input Format:** Value of n **Output Format:** Sum of reciprocals Boundary Conditions: n>0 **Input:** 3 $END 4 **Output:** 11 3.02 $END 31 4.03		
5. **Module:3 Collections and Data Handling in Python** Lists: Create, Access, indexing, slicing, operations, built-in functions, List comprehensions, Tuples: Create, Indexing and slicing, Operations on tuples, Dictionary: Create, add, and replace values, Operations, Dictionary comprehensions, Sets: Creation and operations-Strings: Comparison, Formatting, Slicing, Splitting, Stripping, Regular Expressions: Syntax and meta characters, re module, Regex patterns and groups. **List of Experiments (Indicative)** **Government Committee-(M3-CO2)** Government of India has appointed two separate teams with three members each for investigating a case. After forming the two teams with three members each, as a cost-cutting measure, the government decides to merge the two teams in to a single team with five members, by eliminating the member with the least experience among all the six members from the two teams. Use dictionaries to store the two teams and form a new team by concatenating the two teams. The chairperson of the new team is the person with the maximum experience. Design an algorithm and write the subsequent Python code to store the members of the two teams as dictionaries, print the new (concatenated) dictionary with the names in an increasing order of experience, to check whether a given name is present in the new dictionary or not and to print the name of the chairperson of the new team. Note: Assume that there is only one person with least experience in the two dictionaries. **Input format:** Enter the details of the first team Name of the first member Experience (in years) of the first member	6 hours	

… .... Name of the third member Experience (in years) of the third member Enter the details of the second team: Name of the first member Experience of the first member .... Name of the third member Experience of the third member Enter the name to be checked in the new team **Output format:** Exists or Does not Exist New team with experience as key value Name of the person who is having maximum experience [Hint: use pprint function for printing dictionary in sorted order. Syntax for pprint is pprint(dictionary name) Include the line “from pprint import pprint” in the top of your program **Input:** Raju 9 Ramu 10 Rahim 6 John 15 Joseph 16 Jaitly 3 Ram **Output:** Exists {6: 'Rahim', 9: 'Raju', 10: 'Ramu', 15: 'John', 16: 'Joseph'} Joseph	
6. **Module:3 Collections and Data Handling in Python** Lists: Create, Access, indexing, slicing, operations, built-in functions, List comprehensions, Tuples: Create, Indexing and slicing, Operations on tuples, Dictionary: Create, add, and replace values, Operations, Dictionary comprehensions, Sets: Creation and operations-Strings: Comparison, Formatting, Slicing, Splitting, Stripping, Regular Expressions: Syntax and meta characters, re module, Regex patterns and groups. **List of Experiments (Indicative)** **Arrangement of Plants (M3-CO2)** A gardener has the practice of assigning ID to the plants during plantation. One day, he makes a note of the heights of plants in his garden. He writes the height of the plant against the ID of the plant.	6 hours

He then instructs his employee to keep the plants, in ascending order of its height. Design an algorithm and write the Python code to display the list of ID numbers of plants in ascending order of their height. IDs are also numbers. Check for boundary conditions and print 'Invalid input' for wrong input. For example, if there are three trees with IDs as 175, 160, 120 and height as 47, 73 and 23 then the output should be [120, 175, 160]. **Input Format:** First line contains the number of plants, n Next line contains the ID of the plant-1 Next line contains the height of plant-1 . . . Next line contains the ID of the plant-n Next line contains the height of plantn **Output Format:** IDs sorted according to height. Print one id in one line **Boundary Conditions:** All inputs >=0 Example Input/Output 1: **Input:** 3 175 47 160 73 120 23 $END 2 3 -2 4 **Output:** [120, 175, 160] $END Invalid input	
7.	
Module:3 Collections and Data Handling in Python Lists: Create, Access, indexing, slicing, operations, built-in functions, List c Dictionary comprehensions, Sets: Creation and operations-Strings: Compar groups. **List of Experiments (Indicative)** **Password Check (M3-CO2)** Given a word check if it is a valid password by writing a regular expression A password is said to be 'Good' if it is formed by the following rules: i) Should begin with a letter ii) Should contain atleast one digit and one special character	4 hours

iii) length of the password should be atleast 8 Print 'Bad' if the given word do not follow the above rules **Input** abcd-2 $END dfg **Output** Good $END Bad	
8. **Module:4 Functions, File Operations, and Exception Handling in Python** Functions: Parameters and Arguments: Positional arguments, Keyword arguments, Parameters with default values, Local and Global scope of variables, Functions with Arbitrary arguments, Recursive Functions, Lambda Function. File handling: File modes and File Operations, File objects. Exception Handling: Understanding Errors and Exceptions, Using try, except, else, and finally Blocks, Raising Exceptions. **List of Experiments (Indicative)** **Ride Fare Calculation for a Taxi Service (M4-CO2)** A taxi company wants to build a simple billing system using Python. The fare depends on the following rules: • A base fare of $4.00 is charged for every ride. • Each kilometer traveled adds $1.50 to the fare. • If the ride occurs during peak hours (7 AM–10 AM or 5 PM–8 PM), there's a 20% surcharge on the total fare. Write a function calculate_fare(distance_km, hour) that: • Takes distance_km (a float) representing the distance of the ride in kilometers. • Takes hour (an integer from 0 to 23) representing the hour of the day the ride starts. • Returns the total fare rounded to 2 decimal places. **Example:** calculate_fare(10, 9) # Output: 21.6	6 hours
9. **Module:4 Functions, File Operations, and Exception Handling in Python** Functions: Parameters and Arguments: Positional arguments, Keyword arguments, Parameters with default values, Local and Global scope of variables, Functions with Arbitrary arguments, Recursive Functions, Lambda Function. File handling: File modes and File Operations, File objects. Exception Handling: Understanding Errors and Exceptions, Using try, except, else, and finally Blocks, Raising Exceptions. **List of Experiments (Indicative)** **Student Record (M4-CO3)**	6 hours

You are working for a school that stores student grades in a text file called grades.txt. Each line in the file contains the student's name and their score, separated by a comma Alice,85 Bob,72 Charlie,90 Diana,66 Evan,48 Write a Python function called generate_report(filename) that: 1. Reads the file line by line. 2. Calculates: o The average score of the class. o The names of students who scored below 60. 3. Prints the results in the following format: Average Score: 72.2 Students who need improvement: - Evan	
10. **Module:4 Functions, File Operations, and Exception Handling in Python** Functions: Parameters and Arguments: Positional arguments, Keyword arguments, Parameters with default values, Local and Global scope of variables, Functions with Arbitrary arguments, Recursive Functions, Lambda Function. File handling: File modes and File Operations, File objects. Exception Handling: Understanding Errors and Exceptions, Using try, except, else, and finally Blocks, Raising Exceptions. **List of Experiments (Indicative)** **ATM Withdrawal Simulation (M4-CO3)** You're developing a simplified ATM simulation. Users can attempt to withdraw money from their account using the withdraw(amount) function. However, several issues may arise: • The withdrawal amount may be non-numeric (e.g. user input error). • The withdrawal amount may be greater than the available balance. • The withdrawal amount may be zero or negative, which is not allowed. Write a function withdraw(amount) that: 1. Starts with a predefined balance of $500. 2. Checks if the amount: o Is a valid number (int or float). o Is greater than zero. o Is less than or equal to the balance. 3. Handles exceptions using try-except blocks: o Raises and handles a ValueError for invalid or negative amounts. o Raises and handles a custom exception InsufficientFundsError if the amount exceeds the balance. 4. Prints a message showing: o The withdrawal amount.	4 hours

o The remaining balance. o An error message if the transaction fails. withdraw("abc") # Output: Invalid amount. Please enter a numeric value. withdraw(600) # Output: Insufficient funds. Available balance is $500. withdraw(0) # Output: Withdrawal amount must be greater than zero. withdraw(150) # Output: # Withdrawal successful: $150 # Remaining balance: $350	
11. **Module:5 Data Science Packages** NumPy (Numerical Python): Introduction to NumPy and arrays, Creating and manipulating arrays, Array indexing, slicing, and reshaping, Mathematical operations and broadcasting, NumPy functions and aggregate operations, Pandas (Data Manipulation): Introduction to Series and Data Frame, Creating, reading, and writing data, Indexing, selecting, and filtering data. Handling missing data, Data aggregation and grouping, Merging, joining, and concatenating Data Frames. **List of Experiments (Indicative)** **Student Mark Analysis (M5-CO3)** A teacher wants to analyze marks obtained by 5 students in 3 subjects: Math, Science, and English. The data is represented as a 2D NumPy array: [85, 78, 92], # Student 1 [76, 85, 83], # Student 2 [90, 88, 91], # Student 3 [69, 72, 70], # Student 4 [80, 79, 85] # Student 5 Write a function analyze_marks(marks) that uses NumPy array operations to: 1. Calculate the average marks for each student (row-wise). 2. Calculate the average marks for each subject (column-wise). 3. Find the student with the highest total marks. 4. Return all the above as a dictionary. **Output:** 'student_averages': [85.0, 81.33, 89.67, 70.33, 81.33], 'subject_averages': [80.0, 80.4, 84.2], 'top_student_index': 2	5 hours
12. **Module:5 Data Science Packages** NumPy (Numerical Python): Introduction to NumPy and arrays, Creating and manipulating arrays, Array indexing, slicing, and	5 hours

reshaping, Mathematical operations and broadcasting, NumPy functions and aggregate operations, Pandas (Data Manipulation): Introduction to Series and Data Frame, Creating, reading, and writing data, Indexing, selecting, and filtering data. Handling missing data, Data aggregation and grouping, Merging, joining, and concatenating Data Frames. **List of Experiments (Indicative)** **Sales Data Analysis: (M5-CO3)** You're given a CSV file named sales.csv that contains monthly sales data for multiple salespeople: salesperson,region,month,sales Alice,North,January,1200 Bob,South,January,950 Alice,North,February,1300 Bob,South,February,1100 Charlie,East,January,1000 Charlie,East,February,1050 Alice,North,March,1250 Bob,South,March,1150 Charlie,East,March,1100 Write a function analyze_sales(filename) that: 1. Reads the CSV file into a Pandas DataFrame. 2. Groups the data by salesperson, and calculates: o Total sales per salesperson. o Average monthly sales per salesperson. 3. Groups the data by region, and calculates: o Total sales per region. 4. Returns the results as two DataFrames: one grouped by salesperson and one by region. Output: Salesperson Summary: total_sales average_sales salesperson Alice 3750 1250.0 Bob 3200 1066.67 Charlie 3150 1050.0 Region Summary: total_sales region North 3750 South 3200 East 3150	
13. Module:6 Contemporary Issues	2 hours
Total Laboratory Hours:	**60 hours**
Text Book(s)	
Eric Matthes, " **Python Crash Course: A Hands-On, Project-Based Introduction to Programming,**", No Starch Press American publisher, 3rd Edition, **2023**	

Martin C. Brown, "**Python: The Complete Reference**", McGraw Hill Education, 4th Edition, **2018**	
Reference Books	
Kriti Kumari Sinha, "**Advanced Python Guide: Master Concepts, Build Applications, and Prepare for Interviews**", BPB Publications, 1st Edition, **2024** Charles R. Severance, "**Python for Everybody**", Shroff Publishers, 1st Edition, **2024**	
Mode of Evaluation : Lab Continuous Assessment, Lab Final Assessment, Oral Examination	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
IACSE102	Problem Solving Using Java	0	0	4	2
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

To introduce the fundamental features of the Java programming language and establish a strong foundation in core programming concepts.
To impart skills on Object-Oriented Programming concepts for designing and implementing effective Java-based solutions.
To enhance programming and debugging skills and strengthen the ability to solve real-world problems using Java.

Course Outcomes

Apply fundamental programming constructs and Object-Oriented Programming concepts in Java to develop structured and maintainable code.
Implement inheritance, interfaces, and polymorphism to develop flexible and reusable Java class hierarchies based on problem requirements.
Design robust Java applications by integrating exception handling, multithreading, file I/O, and the Java Collections Framework to solve real-world problems.

Indicative Experiments

1. Banking System : A banking system in Java that allows users to open accounts, deposit money, withdraw money, and calculate interest in savings. The system should handle different types of accounts, such as savings accounts and fixed deposit accounts. Each account type has different interest rates and rules for withdrawals. We need to implement the system in such a way that it handles the deposit and withdrawal of money, calculates interest based on account type, and displays account details in a properly formatted output. Input Format: The first line contains a string representing the account type ("Savings" or "Fixed Deposit"). The second line contains a double representing the initial balance in the account. The third line contains a double representation of the annual interest rate (in percentage) for the account. The fourth line contains an integer indicating the number of months for which the interest should be calculated (for Fixed Deposit only). The fifth line contains a string: "deposit" or "withdraw" depending on the action. If the action is "deposit", the next line will contain a double representing the amount to deposit. If the action is "withdraw", the next line will contain a double representing the amount to withdraw.	4 hours

Output Format: Display the account type. Display the balance after the deposit or withdrawal action, formatted to two decimal places. Display the interest earned based on the account type: For Savings account, calculate the interest based on the balance after the transaction for the year (annual interest rate). For Fixed Deposit account, calculate the interest for the given number of months. Display the total balance after adding the interest (formatted to two decimal places). If a withdrawal is attempted with an insufficient balance, display an error message: "Insufficient funds for withdrawal". Interest Calculation: For Savings Account: Interest=Balance×Rate / 100 For Fixed Deposit Account: Interest=Balance×Rate×Months / 1200 The fixed deposit account earns interest based on the balance and the number of months.	
2. Favourite Number Joe has a favourite number, let's call it X. He wants to check if X is divisible by the sum of its digits. If it is, he considers it a lucky number. If not, he wants to find the closest smaller number, that is divisible by the sum of digits of X. Joe has challenged his friends to solve this puzzle at his birthday party. Input Format The input consists of an integer X, representing Joe's favourite number. Output Format If X is a lucky number, then the output must be in the format: "X is divisible by the sum of its digits." If not, then the output must be in the format: "X is not divisible by the sum of its digits. The closest smaller number that is divisible: Y", where X is the entered number and Y is the closest number.	4 hours
3. Finding Kth Largest Score Imagine a sports event where participants are ranked based on their performance, and medals are awarded to the best athletes. In this scenario, you have an array of athletes' performance scores and you want to identify the kth largest score among all the scores. Input Format The first line of input consists of an integer N, representing the number of athletes The second line consists of N space-separated integers, representing the athlete scores. The third line consists of an integer k. Output Format The output prints the kth largest score.	2 hours
4. Magical Squaring Machine Alex is a young inventor who dreamed of creating a magical machine. This machine could take a series of numbers, perform a special mathematical trick, and reveal their transformed values in terms of square of each element using a lambda expression.	2 hours

Input Format The first line of input consists of an integer N, representing the size of the array. The second line consists of N space-separated integers, representing the element of the array. Output Format The output prints the squares of the elements in the input array, separated by space.	
5. Energy Consumption Tracking Arun wants to create a Java class named Appliance to track the energy consumption of household appliances. Implement methods to input the wattage and daily usage hours, and calculate both daily and monthly energy consumption. Use the main method to create an object of Appliance, gather input from the user, and display the energy consumption, rounded off to 2 decimal places. Your task is to guide him in the program. Note: Daily Consumption = (wattage * hours used) / 1000. Assume there are 30 days in a month. Input Format The first line of input consists of a double value, representing the wattage. The second line consists of a double value, representing the hours used per day. Output Format The first line of output prints "Daily Consumption: X kWh" where X is a double value, rounded off to two decimal places. The second line prints "Monthly Consumption: Y kWh" where Y is a double value, rounded off to two decimal places.	4 hours
6. Duration to Time Conversion James is working on a project to convert seconds to a time format. He wants to create a program that accepts a time duration in seconds and converts it into hours, minutes, and seconds. Help him write a class SecondsToTime with a constructor that gets input in seconds and converts it into hh:mm:ss format. Input Format The input consists of a single integer, representing the time duration in seconds. Output Format The output prints the converted time in the format 'hh:mm:ss' where hh represents hours, mm represents minutes, and ss represents seconds.	4 hours
7. Temperature Monitoring Ravi wants to create a Java class named TemperatureMonitor to track the temperature variations over a period of time. He needs to input the temperature readings for each day and calculate the average temperature for the week. Additionally, he wants to determine the highest and lowest temperatures recorded. Use the this keyword to refer to instance variables and the static keyword to calculate and store the total number of temperature readings across all instances. Ravi wants to	4 hours

be able to create multiple TemperatureMonitor objects for different weeks and print the average, highest, and lowest temperatures for each. Input Format: • The first line contains an integer, representing the number of days (7 for a week). • The next 7 lines contain the temperature readings (as double values) for each day of the week. Output Format: • Print the average temperature for the week, rounded off to two decimal places. • Print the highest and lowest temperatures for the week, rounded off to two decimal places.	
8. Compute Bonus Shasha, an HR manager in a multinational corporation, requires a program to compute bonuses for developers and designers using hierarchical inheritance. Three classes: Employee with a base salary attribute; Developer and Designer extending Employee, each with a bonus percentage, work hours, and methods to calculate and display the final income. Shasha inputs the base salary and work hours for a developer and a designer, and the program outputs their respective final incomes. Note: For developers, the bonus percentage is 10%, for designers, it is 5% Formula: Final income = Base salary + (base salary * bonus percentage * work hours / 100) Input Format The first line of input consists of the base salary (double) and working hours (int) of the developer as space-separated values. The second line consists of the base salary (double) and working hours (int) of the designer as space-separated values. Output Format The output prints the final income (double values) for the developer and designer, respectively, each formatted to two decimal places.	4 hours
9. Resource Modelling Imagine Maria is developing a game, where it involves collecting resources on each planet, and the availability of resources is modelled using a geometric progression. Create an abstract class Series with an abstract method nextTerm(). Implement a subclass called GeometricSeries that calculates the next term in a geometric progression for the resource collection on planets. Allow players to input the initial resource level, resource growth ratio, and the number of planets they plan to explore. Display the predicted resource levels on each planet. Input Format The first line of input consists of an integer, representing the initial resource level. The second line consists of an integer, representing the growth ratio. The third line consists of an integer, representing the number of planets.	4 hours

Output Format The output displays the predicted resource levels of each planet, separated by space. Sample Input: 2 3 5 Sample Output: 2 6 18 54 162	
10. Ticket Price Raj is a software developer tasked with creating a train ticket pricing system for a local railway company using method overriding. The system needs to consider different pricing rules for regular passengers and senior citizens. Senior citizens, aged 60 and above, are eligible for a 10% discount on the regular ticket price. The calculatePrice method in the SeniorCitizenTicket class overrides the same method in the Ticket class to provide a different implementation for senior citizens. Formula: regular price = distance * 0.05, for senior discounted price = regular price * 0.9 Input Format: The input consists of two space-separated integers, representing the distance and age respectively. Output Format: The output prints a double value, representing the ticket price by specifying whether it is a regular or senior citizen ticket. Round off the price to one decimal place.	4 hours
11. Weeks Finding Alex needs to plan a series of events and wants to determine how many complete weeks lie between two important dates. Write a program for him that calculates the number of complete weeks between two given dates, provided in the format yyyy-mm-dd. Use java.time class to handle date parsing and calculations accurately. Input Format The first line of input consists of the start date as a string, in the format 'yyyy mm-dd'. The second line consists of the end date as a string, in the format 'yyyy-mm dd'. Output Format The output prints "X weeks", where X represents the number of complete weeks between the given dates.	4 hours
12. Exception Handling Alice is tasked with creating a program that validates user input for time values (hours, minutes, and seconds). The program should check whether the input values are within the valid range. If any value falls outside the valid range, a custom exception, InvalidHourException, InvalidMinuteException, or InvalidSecondException, should be thrown, indicating which part of the time (hours, minutes, or seconds) is invalid. If no exceptions occur, print "Correct Time - " followed by the time in the format "hours:minutes:seconds." Input Format The first line of input consists of a positive integer, H representing the hours. The second line of input consists of a positive	3 hours

integer, M representing the minutes. The third line of input consists of a positive integer, S representing the seconds. Output Format If the input is valid, the output prints "Correct Time - HH:MM:SS" where H, M, and S are hours, minutes, and seconds respectively. If the hour is invalid, the output prints "Exception occurred: InvalidHourException - hour is greater than 24". If the minute is invalid, the output prints "Exception occurred: InvalidMinuteException - minute is greater than 60". If the second is invalid, the output prints "Exception occurred: InvalidSecondsException - second is greater than 60"	
13. Custom Exception Queency wants to write a program to validate the email address and display suitable exceptions if there is a mistake. Create 3 custom exception classes as below 1. DotException 2. AtTheRateException 3. DomainException A typical email address should have a " . " or "@" character, and the domain name should be valid. Valid domain names for practice are 'in', 'com', 'net', or 'biz'. Display an Invalid Dot usage, Invalid @ usage, or Invalid Domain message based on the email ID. Get the email address from the user, validate the email by checking the above mentioned criteria, and print the validity status of the input email address. Input Format The input consists of a string representing the email to be validated. Output Format If the email is valid, print "Valid email address". If there is a dot exception, print "DotException: Invalid Dot usage", followed by "Invalid email address". If there is a rate exception, print "AtTheRateException: Invalid @ usage", followed by "Invalid email address". If there is a domain exception, print "DomainException: Invalid Domain", followed by "Invalid email address".	3 hours
14. Synchronization Tim is working on a program that alternatively prints even numbers in binary and hexadecimal formats starting from 0. The program has two threads: one for printing numbers in hexadecimal and another for printing numbers in binary. Both threads share a common lock to synchronize their execution. Write a program to accomplish his task. Input Format The input consists of an integer n. Output Format The output prints the number in hexadecimal format and binary format alternatively in the following format: For Hexadecimal: "Hexadecimal: " followed by the hexadecimal value. For Binary: "Binary: " followed by the binary value.	4 hours
15. Char Replacement in a File	3 hours

Cindy is working on a text processing program, and she needs your help. The program is designed to replace all vowels in a given text with a specified consonant. Create a file named file1.txt and prompt the user to input a text. Write the user's input into file1.txt. Then, read the content and perform a specific logic (replacing vowels with a consonant provided by the user), and write the modified result into a new file named file2.txt. Lastly, display the modified result obtained from file2.txt. Input Format The first line of input consists of a string, representing the text that needs vowel replacement. The second line is a single character value, representing the consonant to replace the vowels with. Output Format The output displays the string representing the modified text after replacing the vowels with the specified consonant.	
16. 1) Inventory Items using ArrayList Mesa, a store manager, needs a program to manage inventory items. Define a class ItemType with private attributes for name, deposit, and cost per day. Create an ArrayList in the Main class to store ItemType objects, allowing input and display. Note: Use "%-20s%-20s%-20s" for formatting output in tabular format, display double values with 1 decimal place. Input Format The first line of input is an integer n representing the number of items. For each of the n items, there are three lines: 1. The name of the item (a string) 2. The deposit amount (a double) 3. The cost per day (a double) Output Format The output displays a formatted table with columns for name, deposit and cost per day. 2) Alex is working on a system to store and retrieve employee information for his company's HR department. The system should serialize employee data into a file and deserialize it back when needed. The data should be stored in the form of employee objects, which should contain details like employee ID, name, and salary. By using serialization and deserialization, Alex can ensure that employee data can be saved and later restored, even after the application is closed. Your task is to implement a program that: 1. Serializes employee objects into a file. 2. Deserializes the employee objects from the file. Input Format First line representing N number of employees. Based on that it should accept Employee ID,Employee Name and Employee Salary respectively for all N. Output Format Employees serialized successfully. Displays all Deserialized Employees:	7 hours

3) Sales Analysis using Stream API Peter is working on a project where he needs to analyze sales data for his company. The company records transactions in the form of sales, which include an amount and a category. Peter needs to calculate various statistics from this data, such as total sales, average sales, and sales by category. He needs to use the Stream API to filter, map, and reduce the sales data efficiently. You are tasked with implementing a program that: 1. Takes sales transaction data as input. 2. Filters transactions to find those above a certain threshold. 3. Calculates the total sales, average sales, and sales by category. 4. Uses Stream API operations such as filter, map, and reduce. Input Format: The first line of input will be an integer n, representing the number of sales transactions the user will input. For each transaction, the user will provide the following details: Transaction ID (String) Amount (Double), Category (String) Each of the following n lines will contain these three values separated by spaces. Output Format: The program will print the following outputs: The program will print all transactions where the amount is greater than 500. The program will calculate and print the total sales (sum of all transaction amounts). The program will calculate and print the average sales value. The program will print the total sales for each product category (grouped by category). The program will print all transactions with amounts greater than 500, in string format.	
Total Laboratory Hours:	**60 hours**
Text Book(s)	
Y. Daniel Liang, "**Introduction to Java programming**", Pearson publisher, 11th Edition, **2017** Herbert Schildt, Danny Coward, "**Java: The Complete Reference**"", Tata McGraw Hill, 13th Edition, **2024**	
Mode of Evaluation : Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
IACSE201	Structured and Object-Oriented Programming	2	0	4	4
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

To provide knowledge of the fundamental principles of structured oriented and object- oriented programming paradigms in problem solving through its supported programming languages.
To impart knowledge on modular programming and data handling by using arrays, pointers, and structures.
To develop skills in analysing and solving problems efficiently for addressing realworld challenges.

Course Outcomes

Analyze and compare various programming constructs and choose the most suitable one for addressing the specific problem.
Demonstrate the applications of arrays, strings and pointers in problem solving and idealize the role of pointers in dynamic memory management techniques.
Develop programs using user-defined data types like structures, unions and recognize the applications of modular programming.
Experiment with the principles of Object-Oriented Programming (OOP) by developing programs that connect the concepts of objects and classes to real-world entities and promote modular and reusable code.
Apply polymorphism and templates for solving real world problems and use exception handling mechanisms to recover from errors.

Module:1	**C Programming Basics**	**4 hours**

Introduction to Programming Paradigms: Structured vs OOP- Data Types, Constants, Reserved words, Variables, Operators-Operator precedence and Associativity, Expressions, Type conversions, Input/Output statement, Control Structures: Branching and conditional Statements -if, if-else, nested if, if- else ladder, switch, goto statement, Looping Statementsfor, while, do-while- break and continue statement, Enumerations, Typedef, Preprocessor directives

Module:2	**Arrays Strings and Pointers**	**4 hours**

Introduction to Arrays: Declaration, Initialization – One dimensional arrays –Two dimensional arrays , Strings and its operations, Pointers: Declaration and Access of Pointer VariablesPointer Arithmetic- Array manipulation using pointers, Dynamic Memory Allocation

Module:3	**Structures Functions and Files**	**6 hours**

Structures and Unions: Declaration, Initialization, Access of Structure Variables, Arrays within Structure, Array of Structure, Nested Structures, Pointers to

Structure, Unions, User Defined Functions: Declaration, Definition, Call by Value and Call by Reference, Recursive Functions, Structures and Functions, File management - Opening and closing files, sequential and random access to files. Storage Classes: Scope, Visibility and Lifetime of Variables		
Module:4	**Overview of Object oriented Programming**	**8 hours**
Features of OOP, Classes and Objects, inline functions, Functions with Default Arguments, call by reference, this pointer, Constructors and Destructors, Static Data Members, Static Member functions and objects, Friend Functions and Friend Classes, Inheritance and its types, Constructors in Inheritance		
Module:5	**Polymorphism and Generic Programming**	**6 hours**
Function Overloading, Operator Overloading, Dynamic Polymorphism-Virtual Functions, Pure Virtual Functions, Abstract Classes, Function Templates and Class Templates, Standard Template Library (STL)-Containers, Iterators, Algorithm and Functors, Exception handling		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture Hours:	**30 hours**

Text Book(s)

Yashavant Kanetkar, "**Let Us C**", BPB, 19th Edition, **2022**
Bjarne Stroustrup, "**Programming: Principles and Practice Using C++**", AddisonWesley, 3rd Edition, **2024**

Reference Books

Stanley Lippman and Josee Lajoie, "**C++ Primer**", Addison Wesley, 5th Edition, **2012**
H. M. Deitel and P. J. Deitel, "**C: How to program**", Pearson Education, 9th Edition, **2022**
Brian W. Kernighan and Dennis M. Ritchie , "**The C Programming Language**", Prentice Hall, 2nd Edition, **1988**

Indicative Experiments

1. A.Weather Reporter Rahul is building a weather reporter system. He wants the system to give a message based on the temperature in degrees Celsius entered by the user: • If temperature ≥ 40 → "It's extremely hot!" • If temperature ≥ 30 → "It's hot" • If temperature ≥ 20 → "It's warm" • If temperature ≥ 10 → "It's cool" • If temperature < 10 → "It's cold" Write a C program using the else-if ladder to display the correct message. Input Format: • A single integer temp, representing the temperature in degrees Celsius. Output Format: • A single line showing the weather message. Constraints: • -50 ≤ temp ≤ 60 B. Printing Salary Increment Over Years	4 hours

Arjun is an HR analyst at a company. He is asked to calculate the yearly salary increment for an employee over a given number of years. The employee gets a fixed percentage increment on their salary every year. Write a C program using a loop that takes the initial salary, the annual increment rate, and the number of years as input. For each year, calculate the new salary and display it. salary = salary + (salary * increment_rate / 100); Input Format: • The first line contains a double value S, the initial salary. 4 • The second line contains a double value R, the annual increment rate in percent. • The third line contains an integer N, the number of years. Output Format: • Print N lines. • Each line should contain the salary at the end of that year, rounded to two decimal places. Constraints: • 1.00 ≤ S ≤ 1000000.00 • 0.10 ≤ R ≤ 100.00 • 1 ≤ N ≤ 20 C.Calculator with Multiple Operations Naveen is creating a multi-operation calculator for an engineering application. The calculator should perform different operations based on the user's choice: 1. Addition 2. Subtraction 3. Multiplication 4. Division 5. Modulus 6. Power (X^Y) 7. Exit Write a C program using a switch-case that takes two integers and the user's choice, performs the corresponding operation, and displays the result. For power calculation, use a loop or math.h functions. Input Format: • First line: Integer X • Second line: Integer Y • Third line: Integer choice (1 to 7) Output Format: • A single line showing the result of the operation • If the choice is invalid, print "Invalid choice" Constraints: • -1000 ≤ X, Y ≤ 1000 • Division by zero must be handled (if Y = 0 for division or modulus, print "Cannot divide by zero") D. Student Result Processing System Divya is a data operator in a school. She is asked to prepare a system that reads the marks of N students (out of 100) and assigns a grade to each student based on the following criteria: • Marks ≥ 90 → Grade A • Marks ≥ 80 → Grade B • Marks ≥ 70 → Grade C • Marks ≥ 60 → Grade D • Marks < 60 → Grade F She wants to input the number of students, read their marks one by one, and then print the grade for each student Write a C program that accomplishes this. Input Format: • The first line contains an integer N – number of students. • The next N lines each contain one integer – marks of a student. Output Format: • For each student, output the grade in the following format: Student i: Grade X Constraints: • 1 ≤ N ≤ 100 • 0 ≤ marks ≤ 100	
2. A.Employee ID Verification System Meera works in the HR department of a company. She has a list of employee IDs of people who are allowed to enter the restricted area. She wants to build a system where an employee enters their ID number, and the system checks whether the ID is present in the list or not. Write a C program that: • Takes a list of N employee IDs. • Accepts a search ID from the user. • Uses a loop to search the ID in the list	4 hours

(linear search). • Displays "Access Granted" if the ID is found and "Access Denied" if not. Input Format: • The first line contains an integer N – the number of employees. • The second line contains N space-separated integers – the employee IDs. • The third line contains an integer – the ID to search. Output Format: • A single line: either "Access Granted" or "Access Denied" Constraints: • 1 ≤ N ≤ 100 • Each ID is a positive integer between 1000 and 9999 B. Marks Sorting System Anil is developing a system to process marks for a class of students. Each student has marks in 3 subjects. The principal wants to know the total marks of each student and then sort the students in descending order of their total marks. Write a C program that: • Uses a 2D array to store the marks of N students in 3 subjects. • Calculates the total marks for each student. • Sorts the students based on their total marks (highest to lowest). • Displays each student's marks and total after sorting. Input Format: • First line: Integer N – number of students • Next N lines: 3 space-separated integers – marks in 3 subjects Output Format: • N lines showing 3 subject marks and total marks in the following format: Marks: m1 m2 m3 | Total: T 4 Constraints: • 1 ≤ N ≤ 100 • 0 ≤ marks ≤ 100 for each subject	
3. A. Divya is developing a name processing tool for her school database. She wants to process full names of students and perform the following tasks: Task: Write a C program that: • Accepts the first name and last name of two students. • Combines first and last names into a full name for both students. • Calculates the length of each full name. • Compares the two full names. • Displays: o Full names o Lengths o Whether both names are same or not Input Format: • First name and last name of Student 1 (strings, no spaces) • First name and last name of Student 2 (strings, no spaces) Output Format: • Full Name 1: ... • Full Name 2: ... • Length of Full Name 1: ... • Length of Full Name 2: ... • Names are same / Names are different Constraints: • Names must be alphabetic (only letters) • Each name ≤ 25 characters • Maximum full name length = 50 characters B Priya is developing a program to manage the names of students in a class. She wants to: • Store the names of N students. • Sort the names alphabetically. • Print the sorted list of student names. Task: Write a C program that: 1. Reads the number of students N. 2. Uses a 2D character array (array of strings) to store the names of N students. 3. Sorts the names in alphabetical order (A to Z). 4. Prints the sorted names. Input Format: • First line: Integer N — number of students (1 ≤ N ≤ 100) • Next N lines: Each line contains a student's name (max length 50 characters, no spaces). Output Format: • N lines containing the sorted student names.	4 hours
4. A.Akhil is building a program to help his friends analyse their test scores. Each friend has a set of 5 subject marks. Akhil wants to: 4 1.	4 hours

Store the marks of all students in a 1D array. 2. Use pointers to access and display marks. 3. Calculate the average of the marks using pointers. 4. Display the marks and the average. Task: Write a C program that: • Accepts 5 integer marks from the user. • Uses a pointer to: o Display all marks. o Calculate and print the average. Input Format: 5 space-separated integers (marks in 5 subjects) Output Format: • Marks: m1 m2 m3 m4 m5 • Average: XX.XX Constraints: • Each mark must be between 0 and 100. • Only the main() function should be used. B. Nisha is writing a program to store marks of N students dynamically because the number of students can vary each time the program runs. The marks of each student in 3 subjects will be inputted and stored dynamically. Nisha wants to: • Allocate memory dynamically for storing marks. • Calculate and display the total marks for each student. Task: Write a C program that: • Takes the number of students N as input. • Dynamically allocates memory to store marks for N students, each having 3 subjects. • Reads marks for each student. • Calculates total marks for each student. • Displays the marks and total marks for each student. • Frees the allocated memory at the end. Input Format: • First line: Integer N (number of students) • Next N lines: 3 space-separated integers (marks in 3 subjects) Output Format: For each student, output their marks and total in the format: Student i: Marks = m1 m2 m3 Total = T Constraints: • 1 ≤ N ≤ 100 • 0 ≤ marks ≤ 100	
5. A. A library management system stores details about books. Each book has: • A book ID (integer). • A title (string). • Information about the author, stored in a nested structure: o Author's name (string) o Author's age (integer) 4 • A union to store either: o The price of the book (float), or o The number of copies available (integer) The system should be able to: • Input the book details. • Input whether to store price or number of copies. • Display all the stored information. Input Format: • Book ID (int) • Title (string, max 100 chars, no spaces) • Author name (string, max 50 chars, no spaces) • Author age (int) • Choice p or c (price or copies) • Depending on choice: o Price (float) OR o Copies (int) Output Format: • Display all details in a neat format. Constraints: • Number of books: 1 ≤ N ≤ 100 (if extending to multiple books) • Book ID: positive integer (1 ≤ book ID ≤ 10,000) • Title length: maximum 100 characters, no spaces • Author name length: maximum 50 characters, no spaces • Author age: 18 ≤ age ≤ 100 • Price: 0.01 ≤ price ≤ 10,000.00 • Copies: 0 ≤ copies ≤ 1000 • Choice input: only p or c B. An e-commerce platform needs to store product information. Some products like electronics have warranty (in years), and others like groceries have expiry (in days). Rahul decides to use a union within a structure to manage this efficiently. Write a C program that reads details for a product including ID, name, and type. Based on the type, either warranty or expiry should be stored using a union.	4 hours

Input Format • An integer product_id • A string product_name • A character type ('E' for electronics, 'G' for groceries) • An integer (warranty years for 'E', expiry days for 'G') Output Format A message displaying the product name, ID, and either warranty or expiry information based on type. Constraints • 1 ≤ product_id ≤ 1000 • product_name ≤ 20 characters • Warranty: 1–5 years • Expiry: 1–30 days	
6. A. A bank requires a utility to calculate the compound interest for various accounts. Rahul needs to write a function that calculates and returns the compound interest using the formula: !" = $ ∗ &1 +) 100+ ! – - Write a C program using a user-defined function to calculate compound interest. Input Format • Float P – Principal amount • Float R – Annual rate of interest • Integer T – Time in years Output Format A single line displaying the calculated compound interest Constraints • 1000 ≤ P ≤ 10000 • 1 ≤ R ≤ 15 • 1 ≤ T ≤ 20 B. A university wants to store student data including their name, roll number, and marks in three subjects. It also wants a report that returns the average using a function that takes a structure as an argument. Write a C program that defines a student structure, reads data, and calculates the average using a function. Input Format • String name • Integer roll_number • Three integers for marks Output Format A message showing student name, roll number, and average marks Constraints 0 ≤ marks ≤ 100 C. Rahul is developing a C program to process the ages of students in a class. The ages are stored in a 1D array. Rahul wants to write a function that uses pointers to: • Calculate the average age of all students. • Update the ages by adding 1 year to each student's age (to simulate aging after a year). Task: 1. Read the number of students N. 2. Use a 1D array to store the ages. 3. Write a function that: • Takes a pointer to the first element of the array and the number of students. • Calculates and returns the average age. • Updates the array by incrementing each student's age by 1. 4. In main(), call the function and then print the updated ages and the average age before incrementing. 4 Input Format: • First line: Integer N (number of students) • Second line: N space-separated integers (ages of the students) Output Format: • Print the average age (before incrementing) as a float with 2 decimal places. • Print the updated ages after incrementing.	4 hours
7. Meera is building a small application to store and read student records using files. Each student has a name and total marks. Meera wants the program to: • Accept details for N students. • Write them into a file named students.txt. • Read the file and display the contents in a formatted way. Task: Write a C program that: 1. Takes N (number of students) as input. 2. Accepts each student's name (string) and total_marks (integer). 3. Stores this data into a file named students.txt. 4. Then reads from the file and displays the student records in this format: Name: , Total Marks:	2 hours

Input Format: • First line: Integer N • Next N lines: Each line contains name (no spaces) and total_marks (int) Output Format: --- Student Records --- Name: , Total Marks: Name: , Total Marks: … Constraints: • 1 ≤ N ≤ 100 • 1 ≤ total_marks ≤ 300 • Name is a string without spaces (max 30 characters)	
8. A. Rahul is developing an order system where each Order object should be initialized with an order ID and amount at the time of creation. He decides to use a constructor for this purpose. Write a C++ program using a class with a constructor to initialize and display the order details. Input Format • Integer order_id • Float amount Output Format Order ID and Amount as a formatted output. Constraints • 1 ≤ order_id ≤ 9999 • 0 ≤ amount ≤ 100000 4 B. A student registration system needs to track how many students have been registered. Rahul decides to use a static member in the Student class to keep a count of total students created. Write a C++ program using a class with static data members to count the number of students created. Input Format • Integer n – number of students to register • For each student: name and roll number Output Format Each student's details followed by the total number of students created. Constraints 1 ≤ n ≤ 100	4 hours
9. Nisha is creating a utility billing system where customers are billed based on their electricity usage. She wants to provide a default tax rate of 5% if not specified. If the tax rate is provided, the total bill should include that rate instead. She uses a function with default arguments to implement this. Write a C++ program that takes the number of units consumed and optionally the tax rate, then calculates the total bill using a unit rate of Rs. 10 per unit. Input Format: First line: Units consumed (integer) Second line (optional): Tax rate (float) Output Format: Total bill: Rs Constraints: • Units: 1–1000 • Tax: 0–20%	4 hours
10. A. Sneha is developing a system to compare the bank balances of two customers to determine who has more funds. She decides to use two different classes to represent the customers, but wants to use a single function to access and compare their private balance details. She uses a friend function to achieve this. Write a C++ program using two classes and a friend function that accesses the private balances of both customers and prints who has a higher balance. Input Format: First line: Name and balance of Customer1 Second line: Name and balance of Customer2 Output Format: Customer with higher balance: Constraints: • 1 ≤ balance ≤ 1,000,000 B. Bank – Accessing Private Data Using a Friend Class A bank maintains private account details in a class called BankAccount. These	4 hours

details include: 4 • Account holder name • Balance (private) A separate class BankManager is responsible for: • Viewing account balance • Updating account balance But since the balance is private in BankAccount, the BankManager class cannot access it directly. To solve this, the BankAccount class makes BankManager a friend class, allowing it to access and modify private data directly. Input Format: A string – Account holder's name (no spaces) Line 2: An integer – Initial balance (in ₹) Line 3: An integer – Deposit amount Output Format: Account Holder: Current Balance: Constraints: • 1 ≤ Initial Balance ≤ 1,00,000 • 0 ≤ Deposit Amount ≤ 50,000 • Account holder name has no spaces and length ≤ 20	
11. A.Ravi is designing a system to track employees and their salaries. He creates a base class Employee to store basic details and a derived class Manager that adds a bonus amount to the base salary. He uses inheritance to structure this system efficiently. Write a C++ program using a base class and a derived class to read employee data and display the total salary (base + bonus) for a manager. Input format First line: Name (string) Second line: ID (integer) Third line: Base Salary (float) Fourth line: Bonus (float) Output Format: Manager Name: ID: Total Salary: Rs Constraints: 1000 ≤ ID ≤ 9999 10000 ≤ Salary ≤ 100000 0 ≤ Bonus ≤ 50000 B. An educational institute wants a system to manage students' academic and sports details. • Class Student: stores student info (name, roll_no). • Class Sports: stores sports info (sport_name, score). • Class Result: inherits from both Student and Sports. It also has marks for 3 subjects and calculates the total marks and overall score (academic + sports). You need to create classes using multiple inheritance and display the complete result. Task: 4 • Class Student: o Data members: name (string), roll_no (int) o Member functions: input and display student info • Class Sports: o Data members: sport_name (string), score (int) o Member functions: input and display sports info • Class Result (derived from both Student and Sports): o Data members: marks for 3 subjects (int array) o Member functions: input marks, calculate total marks, display total marks, display overall score (total marks + sports score) Input Format: • Line 1: name (string without spaces) • Line 2: roll_no (int) • Line 3: sport_name (string without spaces) • Line 4: score (int) – sports score • Line 5: 3 space-separated integers – marks in 3 subjects Output Format: Student Details: Name: Roll No: Sport: Sports Score: Academic Marks: m1 m2 m3 Total Academic Marks: Overall Score: Constraints: • roll_no is a positive integer. • Sports score and marks are between 0 and 100. • Name and sport_name are single-word strings without spaces	4 hours
12. Aarav is creating a geometric calculator to find the area of different shapes. He uses function overloading to write different versions of a function area() – one to calculate the area of a square, another for a	4 hours

rectangle, and another for a circle. Write a C++ program that uses function overloading to calculate area based on the input type. Input Format: First line: Choice (1 for square, 2 for rectangle, 3 for circle) Next lines: Appropriate dimension(s) Output Format: Area: value Constraints: • Use 3.14 as the value of π • Side, length, breadth, radius ≤ 1000	

13. A.You are designing a banking system backend. A customer can make multiple transactions. You need to design a Transaction class that supports: • Combining two transactions (+) • Comparing transactions by amount (==,) • Outputting a formatted transaction string (< Compares transactions by amount only << Outputs in format: "[ID] TYPE Amount on TIMESTAMP" If transaction types differ, + should throw std::invalid_argument. Constraints: • type will always be "credit" or "debit" • amount is a positive double with 2 decimal places • timestamp is always valid, and comparisons are lexicographically safe ("YYYY-MM-DD HH:MM") Sample Input & Output Input: Transaction t1(1001, "credit", 100.00, "2025-01-01 08:00"); Transaction t2(1002, "credit", 50.00, "2025-01-01 09:30"); Transaction t3 = t1 + t2; std::cout << (t1 < t2) << std::endl; std::cout << t3 << std::endl; Output: 0 [1003] credit 150.00 on 2025-01-01 09:30 Input with Error (Different Types): Transaction t1(1001, "credit", 100.00, "2025-01-01 08:00"); Transaction t2(1002, "debit", 50.00, "2025-01-01 09:30"); Transaction t3 = t1 + t2; // throws Output: terminate called after throwing an instance of 'std::invalid_argument' what(): Cannot combine credit and debit transactions.
B. Rahul is creating a program to manage time durations in hours, minutes, and seconds. He wants to add two time durations using the + operator and display the total time correctly (carry over seconds to minutes and minutes to hours). Task: Create a Time class that: • Has three integer data members: hours, minutes, and seconds. • Overloads the + operator to add two Time objects. • Correctly handles carry-over: o 60 seconds = 1 minute o 60 minutes = 1 hour • Includes a method to display time in HH:MM:SS format. • Reads input for two times, adds them, and displays the result.
Input Format: Two lines, each containing three integers separated by spaces: h1 m1 s1 h2 m2 s2 • h, m, s represent hours, minutes, seconds respectively.
Output Format: Sum of times: HH:MM:SS
Constraints: • 0 ≤ h < 24 • 0 ≤ m < 60 • 0 ≤ s < 60
	4 hours

14. A. Sonia is building a utility program that needs to find the maximum of two values. But the values could be of any data type integers, floats, or even characters. Instead of writing separate functions for each data type, Sonia wants to write a single function template that works for any type. Task: Write a function template in C++ that: • Takes two parameters of any type. • Returns the greater of	4 hours

the two values. • Demonstrates the function by finding the maximum of: o Two integers o Two floating-point numbers o Two characters For example, if comparing: • 3 and 7 → Output: Max is 7 • 5.5 and 2.3 → Output: Max is 5.5 • 'a' and 'z' → Output: Max is z B.You're working for a data analytics company that processes a wide range of numerical and non-numerical data (integers, floats, doubles, strings, etc.). You are tasked with designing a generic C++ class template that can calculate basic statistics on a dataset of any type. Task: Create a template class Stats that works with any type T (like int, float, double, or even std::string) and supports the following operations: Class Template: Stats Data Member: • std::vector data Public Methods: Method Description void addData(const T& value) Adds a value to the dataset T getMax() const Returns the maximum element T getMin() const Returns the minimum element 4 double getAverage() const Returns the average (only for arithmetic types) size_t size() const Returns the number of elements in the dataset Constraints: • For getAverage(), throw a std::logic_error if T is not an arithmetic type (e.g., std::string). • Use SFINAE or std::enable_if to restrict getAverage() to arithmetic types only. • You must not use hardcoded type-specific logic like if (typeid(T) == ...). Sample Input/Output: Example 1 (Integers): Stats s; s.addData(10); s.addData(20); s.addData(30); std::cout << s.getMin() << " " << s.getMax() << " " << s.getAverage() << std::endl; Output: 10 30 20 Example 2 (Strings): Stats s; s.addData("apple"); s.addData("banana"); s.addData("kiwi"); std::cout << s.getMin() << " " << s.getMax() << std::endl; s.getAverage(); // should throw Output: apple kiwi terminate called after throwing an instance of 'std::logic_error' what(): Average is only supported for arithmetic types.	
15. A.Dynamic Event Registration System (Vector STL) Riya is organizing a tech event and doesn't know in advance how many participants will register. She decides to build a registration system that allows her to: • Dynamically add participant IDs using vector • Display all registered IDs • Search whether a given ID is registered Write a C++ program using the vector STL that: 1. Takes N participant IDs and stores them in a vector 2. Prints the list of IDs 3. Accepts one search ID 4. Searches the ID using a loop and prints "Registered" or "Not Registered" Input Format: 4 • First line: Integer N – number of participants • Second line: N space-separated integers – the participant IDs • Third line: Integer – the ID to search Output Format: • The list of participant IDs in a single line • Then print: "Registered" or "Not Registered" Constraints: • 1 ≤ N ≤ 100 • 1000 ≤ ID ≤ 9999 B. Amit is building a simple Task Manager application where tasks are added, removed, and displayed. The task list should maintain the order in which tasks were added. He decides to use the list STL because: • Tasks need to be added or removed frequently. • Order of tasks must be preserved. Write a C++ program using the list STL to: 1. Add	4 hours

multiple tasks (strings) into a list 2. Display the task list 3. Remove a specific task by name 4. Display the updated list Input Format: • First line: Integer N – number of tasks • Next N lines: Each line contains one task (as a string) • Next line: A string – task to be removed Output Format: • First list of tasks • Updated list after removing the specified task Constraints: • 1 ≤ N ≤ 50 • Task names are non-empty strings without spaces C. You are given a list of Student objects (name, marks). Store them in a set sorted by: 1. Highest marks 2. Lexicographical order of name (if tie) Use a functor or lambda comparator	
16. A.ATM Withdrawal System Rohan is developing an ATM system where users can withdraw money from their bank account. The bank has the following rules: • A user cannot withdraw more money than their current balance. • The withdrawal amount must be positive. Rohan wants to handle these invalid conditions using C++ exception handling (try, throw, catch). Write a C++ program that: 1. Takes the user's current balance as input. 2. Asks the user for an amount to withdraw. 3. If the amount is negative or more than the balance, throw an exception. 2 4. Otherwise, deduct the amount and print the remaining balance. 5. Use try, throw, and catch blocks to handle errors. Input Format: • First line: Integer – current balance • Second line: Integer – withdrawal amount Output Format: • If successful: Withdrawal successful. Remaining balance: • If error: print error message using catch Constraints: • 1 ≤ balance ≤ 1,00,000 B. Student Marks Validation System A college system collects marks for 5 subjects from students. Each mark must be: • Between 0 and 100 • If any subject mark is out of range, the system must throw an exception and display an error. The system should: • Take marks as input for 5 subjects • Validate each mark • If all are valid, display total and average • If any mark is invalid, throw and handle an exception with a proper error message. Input Format: • Five integers (marks for 5 subjects) Output Format: • If all marks are valid: Total = , Average = • If any mark is invalid: Error: Invalid mark entered! All marks must be between 0 and 100. C. You are building the backend for a secure ATM banking system. The system should allow a user to withdraw money from their account. Your task is to implement an Account class that supports: • Withdrawals (with custom exceptions for insufficient funds or invalid amounts) • Deposit operations • Displaying current balance You are required to use custom exception classes to handle error scenarios gracefully. Requirements: 1. Class: Account • Data Members: int accountNumber o std::string accountHolder o double balance • Public Methods: o void deposit(double amount)	2 hours

Adds money to the account. Amount must be positive. § Throw InvalidAmountException if the amount ≤ 0. o void withdraw(double amount) § Subtracts money from the account. § Throw InsufficientFundsException if balance < amount. § Throw InvalidAmountException if amount ≤ 0. o double getBalance() const Custom Exception Classes: class InvalidAmountException : public std::exception • what() returns "Invalid amount entered." class InsufficientFundsException : public std::exception • what() returns "Insufficient balance for withdrawal." Constraints: • Initial balance is always ≥ 0 • Transactions only involve 2 decimal precision • You must use exception handling (try/catch) and custom exception classes	
Total Laboratory Hours:	**60 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar, Group Discussion, Presentaion	
Recommended by Board of Studies :	23-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	C
IASTS101	Qualitative and Quantitative Skills Practice I	3	0	0	1
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

1. To enhance students' quantitative, reasoning, verbal, and life skills through problem-solving and critical thinking.

Course Outcomes

1. Apply quantitative and logical reasoning techniques such as speed math, data arrangement, and analytical problem-solving to solve aptitude and reasoning-based problems efficiently.
2. Demonstrate effective life skills, including profile building, decision making, critical thinking, and communication skills for personal and professional growth in placement and career contexts.

Module:1	Speed Math	10 hours

Addition and Subtraction of Bigger Numbers - Square and square roots-Cubes and cube roots-Vedic Math - Multiplication Shortcuts -Multiplication of 3 and higher digit numbers-Simplifications - Comparing fractions – Calculation Shortcuts - Divisibility rules

Module:2	Percentages, Simple and Compound Interest	6 hours

Percentage -Increase, Decrease – Dealing with Fractions and Decimal- Simple Interest -Compound Interest - Relation between Simple and Compound Interest

Module:3	Profit and Loss, Partnership	5 hours

Profit and Loss – Partnership- Averages and Weighted Averages- Mixtures and Allegations

Module:4	Data Arrangements, Blood relations, Cryptarithmetic and Attention to detail	10 hours

Linear Arrangement -Circular Arrangement-Multi-dimensional Arrangement- Blood relations - Crypt Arithmetic Type: Addition, Multiplication - Attention to detail

Module:5	Life skills	8 hours

Profile building-Learn how to create a strong personal and professional profile - Identify elements of a resume, portfolio, and LinkedIn profile - Decision Making- Understand decision-making models and steps - Understand group vs individual decisions - Critical thinking- Define critical thinking and its importance - Learn to distinguish facts, assumptions, and opinions- Problem solving - Identify types of problems- Apply creative and analytical problem-solving techniques

Module:6	**Reading Comprehension for placements, Critical Reasoning and Voices**	**6 hours**
Types of questions-Comprehension Strategies- Practice exercise- Critical Reasoning- Active voice -Passive voice		
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Dr R. S. Aggarwal," **Quantitative Aptitude for Competitive Examinations**", Chand & Company Ltd, ISBN 978-9358705010, Revised Edition, **2024-25** M. Tyra, "**Magical Book on Quicker Maths**", BSC Publishing Co. Pvt. Ltd, ISBN 978-8190458924,5th Edition, **2024**		
Reference Books		
Stephen R. Covey," **The 7 Habits of Highly Effective People**", Simon & Schuster, ISBN 978-1982137274, 30th Anniversary Edition, **2020** Stella Cottrell, "**Critical Thinking Skills: Effective Analysis, Argument and Reflection**," Bloomsbury Academic (Palgrave Macmillan), ISBN 978-1350322585, 4th Edition, **2023**		
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test		
Recommended by the Board of Studies:		
Approved by Academic Council:		

Course Code	Course Title	L	T	P	C
IASTS102	Qualitative and Quantitative Skills Practice II	3	0	0	1
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

1. Equip learners with key life skills and problem-solving, while preparing them for career opportunities through training in quantitative aptitude, logical reasoning, verbal ability and resume building.

Course Outcomes

1. Apply time management techniques, write professional emails, and prepare well-structured reports to enhance productivity and communication.
2. Solve quantitative problems involving time and work, probability, and logical reasoning with accuracy and confidence.

Module:1	**Life Skills**	**8 hours**

Basics of Life Skills & Time Management - Time Management Techniques - Overcoming Procrastination & Productivity Hacks - Basics of Email Writing - Email Writing Etiquette (Advanced) - Basics of Report Writing - Writing & Editing a Professional Report - Practical Application and Review

Module:2	**Permutation and Combinations, Probability**	**6 hours**

Fundamentals of Counting Principle, Computation of Permutations – Computation of Combinations – Circular Permutation and Probability

Module:3	**Time and Work**	**5 hours**

Work with different efficiencies - Pipes and Cisterns - Work equivalence - Division of wages - Advanced application problems with complexity in calculation

Module:4	**Number System and Logical Reasoning**	**13 hours**

Power Cycle - Reminder Cycle - Applications of number system - Coding and decoding - Visual Reasoning – Clocks – Calendars - Direction Sense -Cubes

Module:5	**Verbal Ability**	**7 hours**

Sentence Correction - Subject Verb agreement – Modifiers – Parallelism - Pronoun Antecedent Agreement - Verb Time sequence – Comparisons – Prepositions - Determiners

Module:6	**Recruitment Essentials**	**6 hours**
Looking at an engineering career through the prism of an effective resume - Importance of a resume - the footprint of a person's career achievements - Designing an effective resume - An effective resume vs. a poor resume - Skills you must build starting today, the requisite? - How does one build skills? Impression Management Getting it right for the interview: Grooming, Body Language and other non-verbal communication skills.		
	Total Lecture Hours:	**45 hours**
Text Book(s)		
R.S. Aggarwal, **" Quantitative Aptitude for Competitive Examinations"**, S. Chand Publishing, ISBN 978-8121924624, Revised Edition, **2024-25** John Sonmez, **" Soft Skills: The Software Developer's Life Manual "**, Manning Publications, ISBN 978-1617292392, 1st Edition, 2014		
Reference Books		
R.S. Aggarwal., **" A Modern Approach to Logical Reasoning "**, S. Chand Publishing, ISBN 978-9352833227, 1st Edition, **2018** Richard Paul & Linda Elder, **" Critical Thinking: Tools for Taking Charge of Your Professional and Personal Life "**, Pearson ISBN 978-0132180915, 2nd Edition, **2014**		
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test		
Recommended by the Board of Studies:		
Approved by Academic Council:		

Course Code	Course Title	L	T	P	C
IASTS201	Qualitative and Quantitative Skills Practice III	3	0	0	1
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives

1. To enhance students' quantitative ability, logical reasoning, verbal ability, and life skills.

Course Outcomes

1. Apply quantitative, reasoning, and verbal ability skills to solve aptitude, data interpretation, and verbal problems accurately and efficiently.
2. Demonstrate life skills, including stress management, adaptability, professional networking, critical thinking, and algorithmic problem-solving for personal and professional growth.

Module:1	**Stress Management, Adaptability and Professional Networking**	**8 hours**
Stress Management- Identify causes and effects of stress - Learn practical stress-reduction techniques. Adaptability and Resilience- Understand adaptability in academic and workplace contexts- Learn how to handle change positively and develop resilience. -Professional Networking- LinkedIn Profile Creation- Learn the importance of professional networking for career growth		
Module:2	**Time, Speed, and Distance**	**5 hours**
Basics-Relative Speed-Average Speed-Problems on Trains -Problems on Boats and Streams- Problem on Races		
Module:3	**Data Interpretation and Inequalities**	**5 hours**
Line chart, Pie chart, Tabular and Coded Inequalities		
Module:4	**Logical Connectives and Syllogisms**	**5 hours**
Basic and Advanced problems – Syllogisms with multiple statements		
Module:5	**Essential English Grammar, Vocabulary**	**10 hours**
Synonyms – Antonyms – Analogy-Confusing words - Sentence Completion- Articles and Prepositions – Direct and Indirect speech		
Module:6	**Recruitment Essentials and Think Algo**	**12 hours**

Introduction to Algorithmic Thinking and Problem-solving, thinking with numbers, thinking with visuals and finding the shortest path, Branching and repetitive problems	
Total Lecture Hours:	**45 hours**
Text Book(s)	
Dr R. S. Aggarwal," **Quantitative Aptitude for Competitive Examinations**", Chand & Company Ltd, ISBN 978-9358705010, Revised Edition, **2024-25** S.P. Bakshi." **Objective General English**," Arihant Publications, ISBN 978-9359980959, 3rd Revised Edition, **2024**	
Reference Books	
Stephen R. Covey," **The 7 Habits of Highly Effective People**", Simon & Schuster, ISBN 978-1982137274, 30th Anniversary Edition, **2020** Stella Cottrell, "**Critical Thinking Skills: Effective Analysis, Argument and Reflection**", Bloomsbury Academic (Palgrave Macmillan), ISBN 978-1350322585,4th Edition, **2023**	
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test	
Recommended by the Board of Studies:	
Approved by Academic Council:	

Course Code	Course Title	L	T	P	C
IASTS202	Qualitative and Quantitative Skills Practice IV	3	0	0	1
Pre-requisite	Nil	**Syllabus Version**			
		1			

Course Objectives:

1. Build teamwork, problem-solving, and professional skills, along with quantitative ability, logical reasoning, and verbal ability.

Course Outcomes:

1. Apply teamwork, collaboration, ethics, and problem-solving skills.
2. Demonstrate quantitative, logical, verbal, and interview skills required for effective performance.

Module:1	**Project Management, Teamwork and Collaboration, Ethical awareness**	**8 hours**
Leadership skills and Conflict resolution, Trust and collaboration- Team building skills- Ethics and Integrity		
Module:2	**Ratios and proportions, problems on ages, equations**	**5 hours**
Ratio - Proportion - Variation-Simple Equation -Problem on Ages		
Module:3	**Logarithms, Progression, Geometry, and Mensuration**	**6 hours**
Logarithms, Arithmetic Progression, Geometric Progression, and Harmonic Progression, Geometry and Mensuration		
Module:4	**Logical Reasoning and Puzzles**	**8 hours**
Venn Diagram, Puzzles, Guesstimation, Data Sufficiency		
Module:5	**Verbal Ability**	**6 hours**
Cloze Test, Verbal analogy, Idioms, and Phrasal Verbs		
Module:6	**Recruitment Essentials and Think Algo**	**12 hours**
Personal Interview -Orientation, Do's and Don'ts, Types, and Mock Interview and How machines think and Introduction to pseudocodes, Algorithmic thinking through puzzles, Expressing and analysing algorithms		
	Total Lecture Hours:	**45 hours**

Text Book(s)
Dr R. S. Aggarwal," **Quantitative Aptitude for Competitive Examinations**", Chand & Company Ltd, ISBN 978-9358705010, Revised Edition, **2024-25** Meena, K., & V. Ayothi, ***"Corporate Soft Skills"*, Tata McGraw-Hill Education**, ISBN 9780071077936, 1st Edition, 2011.
Reference Books
Rao, M. S., *"Soft Skills – **Enhancing Employability**: Connecting Campus with Corporate"*, I. K. International Publishing House Pvt. Ltd., ISBN 9788190675772, 1st Edition, 2010
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test

Recommended by the Board of Studies:	
Approved by Academic Council:	

Course Code	Course Title	L	T	P	C
IAHUM101	India Studies	1	0	0	1
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To introduce students to the foundations of India's traditional knowledge systems, cultural heritage, and civilizational values.
To develop an understanding of India's scientific, technological, and philosophical traditions, and their relevance to contemporary policy frameworks like NEP 2020.
To provide a conceptual and structural understanding of the Indian Constitution, its principles, governance mechanisms, and contemporary significance.

Course Outcomes

Explain the core concepts of India's traditional knowledge systems, culture, civilization, and their historical significance.
Analyse India's scientific, technological, and philosophical traditions and their influence on the development of the Indian Constitution and governance structures
Evaluate the contemporary relevance of Indian knowledge systems and constitutional provisions in addressing modern societal challenges.

Module:1	**Traditional Knowledge Culture and Civilization**	**3 hours**

Traditional Knowledge: definition, nature, characteristics, scope and importance - Kinds of traditional knowledge: Vedas, Upanishads, Śāstras - Traditional knowledge vs. Indigenous knowledge vs. Western knowledge - Introduction to culture and civilization -Features of Indian culture.

Module:2	**Science Technology Knowledge Traditions in India**	**3 hours**

Scientific and technological developments in Ancient, Medieval, and Modern India - Importance of biodiversity in India - *Arthashastra*, *Sukraniti*, *Kamandaka Niti* - Contemporary need for incorporating Indian knowledge systems (IKS) - Indian Knowledge System in NEP 2020

Module:3	**Inreoduction to the Indian Constitution**	**3 hours**

Meaning, nature, and significance of the Constitution - Historical background: Colonial period, freedom struggle & Constituent Assembly - Salient features of the Indian

Constitution - Preamble: Philosophy, ideals, and values - Sources of the Indian Constitution		
Module:4	**Rights Duties Governance Structure**	**3 hours**
Fundamental Rights - Directive Principles of State Policy - Fundamental Duties - Union Government: President, Prime Minister, Parliament - State Government: Governor, Chief Minister, State Legislature - Federalism.		
Module:5	**Constitutional Processes**	**2 hours**
Amendments to the Constitution - Judicial System - Constitutional bodies - Social justice provisions: Reservations, welfare policies.		
Module:6	**Contemporary Issues**	**1 hours**
Debates, Discussions on contemporary issues		
	Total Lecture Hours:	**15 hours**
Text Book(s)		
Shikha Jain, Parul G. Manjul & Somya Joshi , **" Traditional Knowledge Systems and Cultural Heritage."**, Aryan Books International, 1st Edition, **2020** Durga Das Basu ., **"Introduction to the Constitution of India "**, LexisNexis, 23th Edition, **2018**		
Reference Books		
Bidyut Chakraborty & Rajendra Kumar Pandey, **"Indian Government and Politics."**, Sage Publications, 1st Edition, **2008** Amit Jha , **"Traditional Knowledge System in India."**, Atlantic Publishers and Distributors Ltd, 1st Edition, **2024** Granville Austin., **"The Indian Constitution: Cornerstone of a Nation"**, Oxford University Press, 2nd Edition, **1999**		
Mode of Evaluation : Quiz, Final Assessment Test		
Recommended by Board of Studies :	03-11-2025	
Approved by Academic Council : No. 80	20-11-2025	

Course Code	Course Title	L	T	P	C
IAHUM101	India Studies	1	0	0	1
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To introduce students to the foundations of India's traditional knowledge systems, cultural heritage, and civilizational values.
To develop an understanding of India's scientific, technological, and philosophical traditions, and their relevance to contemporary policy frameworks like NEP 2020.
To provide a conceptual and structural understanding of the Indian Constitution, its principles, governance mechanisms, and contemporary significance.

Course Outcomes

Explain the core concepts of India's traditional knowledge systems, culture, civilization, and their historical significance.
Analyse India's scientific, technological, and philosophical traditions and their influence on the development of the Indian Constitution and governance structures
Evaluate the contemporary relevance of Indian knowledge systems and constitutional provisions in addressing modern societal challenges.

Module:1	**Traditional Knowledge Culture and Civilization**	**3 hours**

Traditional Knowledge: definition, nature, characteristics, scope and importance - Kinds of traditional knowledge: Vedas, Upanishads, Śāstras - Traditional knowledge vs. Indigenous knowledge vs. Western knowledge - Introduction to culture and civilization -Features of Indian culture.

Module:2	**Science Technology Knowledge Traditions in India**	**3 hours**

Scientific and technological developments in Ancient, Medieval, and Modern India - Importance of biodiversity in India - *Arthashastra*, *Sukraniti*, *Kamandaka Niti* - Contemporary need for incorporating Indian knowledge systems (IKS) - Indian Knowledge System in NEP 2020

Module:3	**Inreoduction to the Indian Constitution**	**3 hours**

Meaning, nature, and significance of the Constitution - Historical background: Colonial period, freedom struggle & Constituent Assembly - Salient features of the Indian

Constitution - Preamble: Philosophy, ideals, and values - Sources of the Indian Constitution		
Module:4	**Rights Duties Governance Structure**	**3 hours**
Fundamental Rights - Directive Principles of State Policy - Fundamental Duties - Union Government: President, Prime Minister, Parliament - State Government: Governor, Chief Minister, State Legislature - Federalism.		
Module:5	**Constitutional Processes**	**2 hours**
Amendments to the Constitution - Judicial System - Constitutional bodies - Social justice provisions: Reservations, welfare policies.		
Module:6	**Contemporary Issues**	**1 hours**
Debates, Discussions on contemporary issues		
	Total Lecture Hours:	**15 hours**
Text Book(s)		
Shikha Jain, Parul G. Manjul & Somya Joshi , **" Traditional Knowledge Systems and Cultural Heritage."**, Aryan Books International, 1st Edition, **2020** Durga Das Basu ., **"Introduction to the Constitution of India "**, LexisNexis, 23th Edition, **2018**		
Reference Books		
Bidyut Chakraborty & Rajendra Kumar Pandey, **"Indian Government and Politics."**, Sage Publications, 1st Edition, **2008** Amit Jha , **"Traditional Knowledge System in India."**, Atlantic Publishers and Distributors Ltd, 1st Edition, **2024** Granville Austin., **"The Indian Constitution: Cornerstone of a Nation"**, Oxford University Press, 2nd Edition, **1999**		
Mode of Evaluation : Quiz, Final Assessment Test		
Recommended by Board of Studies :	03-11-2025	
Approved by Academic Council : No. 80	20-11-2025	

Course Code	Course Title	L	T	P	C
IAHUM109	Principles of Sociology	3	0	0	3
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To introduce the fundamental concepts and principles of sociology
2. To familiarize the students to the basic social processes of society, social institutions and patterns of social behavior.
3. To explore and understand sociology not merely as a social science discipline but as a distinctive branch of knowledge.

Course Outcomes

1. Understand the fundamental concepts, nature, and scope of sociology as a scientific study of society.
2. Analyse the relationship between sociology and other social sciences to understand the interconnections of human behaviour and institutions
3. Remembering the processes of socialization and the role of culture, social control, and social groups in shaping individual and group behaviour
4. Understand the structure and functions of major social institutions such as family, marriage, religion, education, economy, and polity
5. Apply the dynamics of social processesâ€"cooperation, competition, conflict, and the patterns of social change, stratification, and modernization
6. Analyse and prescribe models and solutions to address societal issues

Module:1	**Sociology**	**8 hours**

Definition - Nature - Scope (Formalistic or specialist school & Synthetic school) - Field - Importance - Sociology relationship with Economics, History, Psychology, Anthropology, Political Science.

Module:2	**Sociological Concepts**	**6 hours**

Society, community, association, institution, Social Process, Culture - meaning, characteristics, elements, and importance, Social structure, Status and role.

Module:3	**Socialization and Social Groups**	**9 hours**

Meaning, process, and stages of socialization, Agencies of socialization - family, school, peer group, media, social control - meaning, types (formal and informal), and agencies, social groups - meaning and types (primary, secondary, in-group, out-group, reference group)

Module:4	**Social Institutions**	**9 hours**
Family - types, functions, and changes, Marriage - forms, functions, and contemporary trends, Religion - sociological perspectives, Education - social role and its impact on social mobility, Economy and polity as social institutions		
Module:5	**Social Stratification**	**9 hours**
Social stratification - meaning, forms (caste, class, race), and mobility, social change - meaning, factors, social disorganization - causes and consequences, Modernization and globalization - sociological implications		
Module:6	**Contemporary Issues**	**2 hours**
Guest Lectures		
	Total Lecture Hours:	**43 hours**

Text Book(s)

1. Richard T. Schaefer , "**Sociology**", McGraw Hill., 1st Edition, **2021**
2. C.N. Shankar Rao , "**Sociology : Principles of Sociology with an Introduction to Social Thoughts,**", S Chand & Company Ltd, 8th Edition, **2024**

Reference Books

1. Antony Giddens and Philip W. Sutton (2017), Sociology (8th Edition), Atlantic Publishers & Distributors Pvt. Ltd, "**Sociology** ", Atlantic Publishers Distributors Pvt. Ltd, 1st Edition, **2017**
2. Haralmbos, M. & Holborn (, "**Sociology (8th Edition): Themes and Perspectives,**", Collins Publishers, 1st Edition, **2022**
3. T.B. Bottomore â€“ Sociology: A Guide to Problems and Literature, "**Sociology: A Guide to Problems and Literature**", S.Chand and Company, 1st Edition, **2023**

Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test

Recommended by Board of Studies :	18-05-2026
Approved by Academic Council : No. 82	11-06-2026

Course Code	Course Title	L	T	P	C
IAENG101	Technical English Communication	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

To enable learners to decipher the tenets of general and interpersonal communication
To facilitate application of grammar skills in academic and general situations
To help foster learning of productive skills for communicative competence

Course Outcomes

Enhance writing skills for better publication
Identify the basic concepts related to various aspects of communication
Apply grammatical rules in academic and real-life contexts
Demonstrate analytical reading skills in terms of deciphering research articles
Illustrate qualitative writing skills by drafting emails and academic documents

Module:1	**Introduction to Technical Communication**	**6 hours**

The Nature of Technical Communication

Taxonomy of Technical Writing-Scientific Writing, Technical Reporting

Business Communication

5Cs of Technical Writing
Activity: Applying these concepts to answer conceptual questions

Module:2	**Grammar**	**9 hours**

Concord

Tense

Conditionals

Voice

Transitions and Linking words
Activity: Error Detection

Module:3	**Reading**	**10 hours**

Analysing definitions – Types of definitions, Methods for Expanding Definitions Reading Comprehension – Drawing Inferences from Scientific Texts Structure of a Research Article Analysing a Research Article **Activity:** Reading a research article		
Module:4	**Writing**	**12 hours**
Research Writing: Structure of a Review Paper Types of Reviews Literature Review Referencing Style - IEEE, APA **Activity:** Writing a literature review		
Module:5	**Interpersonal Communication**	**6 hours**
Cross-cultural Communication Interpersonal Communication Leadership Styles Belbin's Team Roles Negotiation Skills Thomas-Kilmann Modes of Conflict Management Statement of Purpose **Activity:** Case study Analysis		
Module:6	**Contemporary Topics**	**2 hours**
Activity: Guest lectures from Industry and, Research and Development Organizations		
	Total Lecture Hours:	**45 hours**
Text Book(s)		
Lannon, J.M. & Gurak, L.J., "**Technical communication**", Pearson, 7th Edition, **2025** Shoba, K.N. & Sam, P., "**Technical English: A workbook**", Cambridge University Press, **2018**		

Reference Books	
Alred, G.J., Brusaw, C.T., & Oliu, W.E., "**Handbook of technical writing**", Bedford St Martins, 11th Edition, **2015** Gross, A., Hamlin, A., Merck, B., Rubio, C., Naas, J., Savage, M., & Desilva, M. , "**Technical writing. "**, Open Oregon Educational Resources, **2017** Laplante, T.A. , "**Technical writing: A practical guide for engineers, scientists, and nontechnical professionals."**, CRC Press, 2nd Edition, **2019** Markel, M. & Selber, S.A. , "**Technical communication."**, Bedford St Martins , 12th Edition, **2018**	
Indicative Experiments	
1. **Vocabulary and Proof Reading** Verbal Traps in Technical Communication: Cliches- Malapropisms- Erroneous Heterographs- The Laziness of "very" Proofreading Symbols Proofreading a Short Passage **Activity**: Interactive Worksheets, Proof Reading	4 hours
2. **Intensive Listening** Listening to Podcasts Viewing Documentaries **Activity**: Listening and Note-taking/ Summarising	8 hours
3. **Speaking** Role Play Storytelling Group Discussion Video Resume Presentations **Activity**: Individual and group presentations	8 hours
4. **Reading** Elements of Fiction-Plot-Character-Setting-Theme-Point of View	6 hours

Appreciation of a text-The Ant and the Grasshopper by Somerset Maugham **Activity**: Story/ Book Review	
5. **Writing** Blog Writing Transcoding Statement of Purpose Creating a Portfolio **Activity**: Blog/ Portfolio Design	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-05-2025
Approved by Academic Council : No. 78	12-06-2025

Course Code	Course Title	L	T	P	J	C
IAENG100	Effective English	0	0	4	0	2
Pre-requisite	NIL	Syllabus version				
		1.0				

Course Objectives

1. To develop fluency in basic communication skills at various academic and Social Contexts.

2. To help students overcome learning difficulties in grammar, vocabulary and pronunciation.

Course Outcomes

Upon completion of the course, learners will be able to:

1. Illustrate listening and reading competence in general and academic contexts
2. Demonstrate sufficient speaking skills in terms of pronunciation and discourse management
3. Apply grammar rules in oral and written communication
4. Employ appropriate vocabulary in diverse situations

Module:1	Listening	5 hours

Listening for General Information, Listening for Specific Information -Listening to formal and informal conversations, listening to talks

Activity: Listening and note-taking

Module:2	Pronunciation	8 hours

Reading Aloud - short and long passages, Listen & Repeat Drills - words with common endings, Tongue Twisters, Mute Words, Commonly Mispronounced Words

Activity: Reading a Passage Aloud

Module:3	Speaking	8 hours

Self-Introduction, Situational Role Play – Formal and Informal Situations, Picture Description - Individual and Group work

Activity: Role Play

Module:4	Grammar	10 hours

Parts of Speech – An overview of the eight parts of speech along with the simple and the progressive tense, Types of Sentences: Affirmative Sentences, Negative Sentences, Interrogative Sentences, and Exclamatory Sentences

Activity: Interactive Worksheets and Pair/Group Activities

Module:5	Vocabulary	8 hours

Synonyms, Antonyms, Homophones, Homonyms, Everyday Vocabulary – Words Describing the Weather, People, and Emotions

Activity: Interactive Worksheets and Pair/Group Activities

Module:6	Reading	10 hours

Activity: Reading Comprehension, Reading Newspaper Articles

Module:7	Writing	11 hours

Activity: Jumbled Sentences, Parts of the Paragraph, Paragraph Writing - Guided and Unguided

	Total Lecture hours:	60 hours

Text Book(s)

1. Chase, B.T. & Johannsen, K.L. (2018). Pathways: Listening, speaking and critical thinking 1. 2nd ed. National Geographic Learning.
2. Douglas, N. & Bohlke, D. (2020). Reading explorer 3. 3rd ed. National Geographic Learning.

3.	Murphy, R. (2024). Essential English grammar. 2nd ed. Cambridge University Press.			
Reference Books				
1.	Elbaum, S.N. & Peman, J.P. (2021). Grammar in context: Basic. 7th ed. National Geographic Learning.			
2.	Folse, K. (2014). Great writing 1: Great sentences for great paragraphs. 4th ed. National Geographic Learning.			
3.	Redman, S. (2017). English vocabulary in use: Pre-intermediate and intermediate. 4th ed. Cambridge University Press.			
4.	Singleton, J. (2005). Writers at work: The paragraph. Cambridge University Press.			
Mode of Assessment: Continuous assessment / FAT / Oral examination and others				
Recommended by Board of Studies		19-02-2025		
Approved by Academic Council		No. 78	Date	12.06.2025

Professional Core Courses

Course Code	Course Title	L	T	P	C
IACSI101	Data Structures and Algorithms	3	0	2	4
Pre-requisite	NIL	Syllabus Version			
		1			

Course Objectives

1. To analyze the time complexity of recursive algorithms.
2. To focus on linear and non-linear data structures and their operations.
3. To provide an insight into graph algorithms.

Course Outcomes

1. Grasp the fundamentals of data structures and calculate the time complexity for a given algorithm
2. Apply operations of linear data structures correctly to problems.
3. Analyze the various searching, sorting and hashing algorithms.
4. Implement all operations in basic and advanced tree structures.
5. Experiment the graph algorithms for impactful real-world problem-solving.

Module:1	Analysis of Algorithms	8 hours

Abstraction - Abstract data types; Data Representation; Elementary data types; Basic concepts of data Structures; analysis of recursive (Towers of Hanoi) and non-recursive algorithm; Asymptotic notations- space and time complexity of algorithm, solving recurrences methods — Iteration, recursive tree, and master method.

Module:2	Linear Data Structures	8 hours

Abstract data type array-1D and 2D array, advantages and disadvantages, Abstract data type stack, Applications of stack — expression evaluation, conversion of expression –infix to prefix, infix to postfix, Queue, types of queue – double ended, circular, applications, List — singly, doubly, circular, applications.

Module:3	Search Sort and Hashing Techniques	10 hours

Searching – Linear search, binary search. Common sorting strategies - bubble sort, insertion sort, selection sort, quick sort, merge sort. Hashing: Hash tables, Hash functions, separate chaining, Open addressing- Linear probing, Quadratic probing, double hashing; extendible hashing.

Module:4	Tree Data Structures	11 hours

General specification and properties of trees, binary tree, tree traversals, expression tree, k-ary, Binary search tree — insertion, deletion, finding min and max element, finding kth element of tree. AVL trees: Terminology, operations - rotation, insertion, and deletion. Heaps- Heap sort- priority queue using heaps, binary, binomial, and Fibonacci heaps.

Module:5	Graphs Algorithms	6 hours

Course Code	Course Title	L	T	P	C
	Complex Variables and Integral Transforms	**3**	**1**	**0**	**4**
Pre-requisite	**Multivariable Calculus and Differential Equations**	**Syllabus version**			

Course Objectives

1. To develop a strong foundation in complex analysis, including analytic functions, and conformal mapping for solving engineering problems in fluid dynamics and electromagnetic fields.
2. To utilize advanced mathematical techniques, such as power series, contour integration and bilinear transformations, to solve complex equations and optimize engineering computations.
3. To apply Laplace and Fourier transforms to analyze and process signals, aiding in system modeling, control theory, and real-time engineering applications.

Course Outcomes

At the end of the course, the student is able to:

1. construct analytic functions and find complex potential of fluid flow and electric field problems.
2. realize the importance of transformations and apply them to identify the boundaries of images.
3. evaluate real integrals using the technique of contour integration.
4. understand basic concepts of Laplace transforms.
5. employ the tools of Fourier transform for solving engineering problems.

Module 1	**Analytic Functions**	**7 hours**

Complex variable- Analytic functions and Cauchy–Riemann equations - Laplace equation and Harmonic functions - Construction of Harmonic conjugate and analytic functions - Applications of analytic functions to fluid-flow and Field problems.

Module 2	**Conformal and Bilinear Transformations**	**9 hours**

Conformal mapping - Elementary transformations - Translation, magnification, rotation, inversion. Exponential and Square transformations ($w = e^z$, z^2) - Bilinear transformation - Cross-ratio- Images of the regions bounded by straight lines under the above transformations.

Module 3	**Power Series and Complex Integration**	**9 hours**

Functions given by Power Series - Taylor and Laurent series -singularities - poles – Residues. Integration of a complex function along a contour - Cauchy-Goursat theorem- Cauchy's integral formula -Cauchy's residue theorem - Evaluation of real integrals -

Module 4	**Laplace transform**	**9 hours**

Definition of Laplace transform – Properties - Laplace transform of periodic functions- Laplace transform of unit step function, Impulse function- Inverse Laplace transforms- Method of partial fractions – Other methods of finding inverse transforms – Convolution theorem.

Module 5	**Fourier transform**	**9 hours**

Complex form of the Fourier transformation- properties – Relation between Laplace and Fourier transforms – Fourier sine and cosine transforms – Convolution theorem and Parseval's identity.

Module 6	**Contemporary Topics**	**2 hours**

	Total Lecture hours: **Total Tutorial hours:**	**45 hours** **15 hours**

Text Book(s)	
1.	G. Dennis Zill, Patrick D. Shanahan, Complex Analysis, 3rd Edition, Jones and Bartlett Publishers, 2019
2	B.S. Grewal, Higher Engineering Mathematics, 44th Edition, Khanna Publishers, 2020
Reference Books	
1.	Erwin Kreyszig, Advanced Engineering Mathematics, 10th Edition, Wiley India, 2015.
2.	B. J. Lewis, E. N. Onder, A.A. Prudil, Advanced Mathematics for Engineering Students – The Essential Toolbox, Elsevier B-H, 2021.

Mode of Evaluation: Quizzes, Assignments, CATs and FAT			
Recommended by Board of Studies	14-May-2025		
Approved by Academic Council		Date	

Representation of Graph, Graph traversal – Breadth First Search (BFS), Depth First Search (DFS), Applications of graphs: minimum spanning tree – Kruskal and Prim, single source shortest path – Dijkstra's algorithm.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. , "**Introduction to Algorithms**", PHI Learning, 4th Edition, **2022**

Reference Books

Yedidyah Langsam, Moshe. Augenstein, Aaron M. Tenenbaum, "**Data Structures Using C and C++**", Pearson Education Inc, 2nd Edition, **2015**
Adam Drozdek, "**Data structures and algorithms in C++**", International Edition, 4th Edition, **2015**

Indicative Experiments

1. Essential Recursive functions - Towers of Hanoi (ToH)	2 hours
2. Implement a queue using two stacks and analyze the running time of the queue operations.	2 hours
3. Implement two stacks in one array A[1 : n] in such a way that neither stack overflows unless the total number of elements in both stacks together is n. The PUSH and POP operations should run in O(1) time. Implement the applications of stack (expression evaluation, conversion of expression).	2 hours
4. The dynamic-set operation UNION takes two disjoint sets S1 and S2 as input, and it returns a set S = S1 ∪ S2 consisting of all the elements of S1 and S2. The sets S1 and S2 are usually destroyed by the operation. Implement the UNION in O(1) time using a suitable list data structure.	2 hours
5. After final exams, a university department prepares a rank list based on student CGPAs. As new results come in, the list must be updated dynamically. Apply an insertion sort algorithm to efficiently place each new student into the correct position in the sorted list.	2 hours
6. Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length m = 11 using open addressing. Illustrate the result of inserting these keys using linear probing with h(k, i) = (k + i) mod m and using double hashing with h1(k) = k and h2(k) = 1 + (k mod (m − 1)).	2 hours
7. A multinational company maintains a structured employee hierarchy where each manager supervises a set of subordinates. This	2 hours

structure is naturally modeled as a tree, with the CEO as the root and each employee as a node. Finding the lowest common manager between two employees and print the hierarchy in preorder/postorder traversal to help HR visualize reporting lines. You may use Tree Traversal: preorder, inorder, postorder to solve this problem.	
8. Sort a given set of nnumbers by first building a binary search tree containing these numbers (usingTREE-INSERT repeatedly to insert the numbers one by one) and then print thenumbers by an inorder tree walk.	2 hours
9. In an operating system that only provides a priority queue for task management, how can you implement both a FIFO queue for managing print jobs and a stack for handling undo operations? Explain how priority values should be assigned in each case to ensure the correct processing order, and discuss why using a priority queue in this way can be beneficial.	2 hours
10. Find your way out of a maze by using Depth First Search (DFS), marking each path with pennies to keep track of the routes you have explored.	4 hours
11. A directed graph G = (V, E) is singly connected if u $\rightsquigarrow$ v implies that G contains at most one simple path from u to v for all vertices u, v $\in$ V. Implement a code to determine whether a directed graph is singly connected.	4 hours
12. In planning a newtransportation network connecting multiple cities, engineers aim to minimizethe total construction cost of roads or railways while ensuring that every cityis reachable from any other city. Design and implement this network usingPrim's algorithm.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation : Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI201	Design and Analysis of Algorithms	3	0	2	4
Pre-requisite	IACSI101	**Syllabus Version**			
		1			

Course Objectives

1. To provide mathematical foundations for analyzing the run time complexity of the algorithms and proofs of its correctness.
2. To apply important algorithmic design paradigms and methods of analysis.
3. To analyze and evaluate algorithms across areas including graphs, strings, and computational hardness theory.

Course Outcomes

1. Understand the proof of correctness for algorithms and analyze its running time.
2. Identify and apply suitable algorithmic design paradigms to solve a given problem.
3. Analyze the performance of major graph algorithms and string-matching algorithms rigorous mathematical techniques.
4. Explore the graph algorithms and string-matching algorithms to solve practical computational problems.
5. Ability to prove the hardness of real-world problems and do performance analysis of randomized and approximation algorithms.

Module:1	**Algorithm Analysis**	**7 hours**

Asymptotic analysis for recurrence relation: Substitution method, Recursion Tree method, Iteration Method, Master method.

Module:2	**Design Paradigms I**	**9 hours**

Brute Force Technique: Finding Maximum and Minimum element in an array, Travelling Salesman Problem - Divide and Conquer: Maximum Subarray Sum, Strassen's Matrix Multiplication - Greedy Technique: Fractional Knapsack Problem, Huffman Coding.

Module:3	**Design Paradigms II**	**10 hours**

Dynamic Programming: Matrix Chain Multiplication, Longest Common Subsequence, 0-1 Knapsack Problem - Backtracking: N-Queens Problem, Subset Sum Problem, Branch & Bound: LIFO and FIFO B&B - 0/1 Knapsack Problem.

Module:4	**String Matching and Graph Algorithms**	**9 hours**

String Matching: NaÃ¯ve String-matching Algorithm, Rabin-Karp Algorithm - Graph

Algorithms:

Shortest Path: Bellman Ford Algorithm, Floyd-Warshall Algorithm - Network Flows: Flow Networks, Maximum Flows: Ford-Fulkerson Algorithm, Edmond- Karp Algorithm.		
Module:5	**Computational Complexity and Approximation Algorithms**	**8 hours**
Complexity classes: P, NP, Reductions, NP-Completeness and NP hard - SAT, Clique, Vertex-Cover - Randomized Algorithms: Las Vegas algorithm: Randomized Quick Sort, Monte Carlo algorithm: Primality Testing - Approximation Algorithm: Vertex Cover, Travelling Salesman Problem.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, "**Introduction to Algorithms**", PHI Learning, 4th Edition, **2022**
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajesekaran, "**Computer Algorithms**", Silicon Press, 2nd Edition, **2008**

Reference Books

1. Anany Levitin, "**Introduction to the Design and Analysis of Algorithms**", Pearson Education Inc, 3rd Edition, **2021**
2. Robert Sedgewick and Kevin Wayne, "**Algorithms**", Pearson Education Inc, 4th Edition, **2021**

Indicative Experiments

1. A stock market analyst wants to find the highest and lowest stock prices from the daily prices of a company over the last N days. Implement the Brute Force technique, implement an algorithm to determine the maximum and minimum stock prices from the givenarray of prices.	2 hours
2. A delivery company wants to minimize the travel cost of its truck that needs to deliver packages to â€˜n' different cities. The cost of traveling between cities is given in the distance matrix D. Implement the Brute Force technique to compute the shortest possible tour that visits each city exactly once and returns to the starting city.	2 hours
3. An e-commerce company records the daily profit/loss values of its online sales for a month (30 days). Some days may show losses (negative values). Using the Divide and Conquer technique implement an algorithm to determine the period (contiguous days) during which the company achieved the maximum profit.	2 hours

4. A graphics rendering software requires multiplying two large nxn transformation matrices for faster image processing. Implement Strassen's Matrix Multiplication algorithm to multiply the two matrices.	2 hours
5. A relief organization is air-dropping essential supplies to a flood-affected region. The aircraft has a maximum carrying capacity of C kg. The supplies are available, each with a specific weight (W1, W2, W3, W4) and value (P1, P2, P3, P4) for the items say Food packets, Medicines, Blankets, Water bottles. Implement the Fractional Knapsack problem to determine the optimal combination of supplies that maximizes the total value while staying within the aircraft's weight limit.	2 hours
6. A text compression software is designed to encode characters for efficient storage. The frequencies of 'n' characters in a file are given as f1, f2, f3, ..., fn. Using Huffman Coding algorithm implement a solution to generate an optimal encoding for these characters based on their frequencies	2 hours
7. In bioinformatics, researchers often compare DNA sequences to find similarities. Given the two DNA sequences: S1 and S2. Design a solution to find the Longest Common Subsequence (LCS) of the two DNA strings using dynamic programming.	2 hours
8. An e-commerce delivery truck has a weight capacity of C units. The company must decide which packages to load to maximize profit. The items, weights and the profits say, n={1, 2, 3, 4}, wi={w1, w2, w3, w4}, and profit={p1, p2, p3, p4}. Write a program to solve the problem using Dynamic Programming.	2 hours
9. A security company is designing a surveillance system for an N Ã– N museum hall where each queen represents a CCTV camera that can monitor its entire row, column, and both diagonals. Model this problem using the N-Queens backtracking algorithm. For N, implement step-by-step how the algorithm places the cameras and backtracks when a conflict arises.	2 hours
10. A disaster-relief team is preparing a rescue helicopter that carry a maximum load of C kg. They have several emergency supply items, each with a weight (in kg) and value (representing importance for survival). Items = {1,2,3,4}, Weights = {w1, w2, w3, w4} and Value = {v1, v2, v3, v4}. Solve this problem using branch and bound algorithm.	2 hours
11. A DNA lab stores genetic sequences in text format. A biologist wants to search whether the substring of length â€œnâ€ occurs in the DNA sequence of length â€œmâ€. Design a solution for the biologist using NaÃ¯ve string-matching algorithm to address his requirement.	2 hours

12. Demonstrate the use of Rabin-Karp algorithm to help a cybersecurity team that scans network packets to detect a malicious signature string â€œnâ€ inside a packet payload given by â€œmâ€ .	2 hours
13. A logistics company wants to calculate the shortest delivery cost from a central warehouse to all its retail outlets. The delivery costs between warehouses and outlets may sometimes be negative due to discounts or reimbursements. Design a solution for	2 hours
this company using Bellman-Ford Algorithm (Single-Source Shortest Path with Negative Weights)	
14. A telecom company wants to place signal boosters at certain network towers to cover all connections between towers, but they want to minimize the number of boosters due to cost constraints. Implement 2-approximation algorithm for vertex cover help in deciding where to place boosters efficiently.	4 hours
Total Laboratory Hours	**30 hours**

Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar, Case Study	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI202	Digital Logic and Computer Design	3	0	2	4
Pre-requisite		Syllabus Version			
		1.0			

Course Objectives

1. Acquaint students with the basic concepts of digital and binary systems
2. Build knowledge on design of combinational and sequential logic circuits for real world applications.
3. Demonstrate theoretical concepts in designing the circuits using appropriate tools and hardware.

Course Outcomes

1. Differentiate the types of number systems and their representations.
2. Express logic functions using Boolean Algebra and K-map simplifications.
3. Design combinational and sequential circuits with optimized logic.
4. Analyze standard combinational circuits such as encoders, decoders, multiplexers, and demultiplexers.
5. Construct arithmetic and logic circuits while incorporating design thinking principles.

Module:1	Digital Logic and Boolean Algebra	9 hours

Number System, Base Conversion, Binary Codes, Complements, Logic gates, Universal gates, Positive and Negative Logic. Boolean algebra, Properties of Boolean algebra, Boolean functions, Canonical and Standard forms, Karnaugh map (up to 5 variables), Dont care conditions, Tabulation Method (up to 5 variables).

Module:2	Combinational Logic Circuits	9 hours

Design of combinational circuits, Adder, Subtractor, Code Converter, Analyzing a Combinational Circuit. Binary Parallel Adder, Magnitude Comparator, Decoders, Encoders, Multiplexers, De-multiplexers.

Module:3	Sequential Logic Circuits	7 hours

Flip Flops, Conversion of Flip flops, Design and Analysis of Sequential circuits

Module:4	Registers and Counters	9 hours

Registers, Shift Registers, Bi-directional shift registers, Counters, Ripple and Synchronous Counters, Ring and Johnson counters.

Module:5	Arithmetic Logic Unit	9 hours

Bus Organization, ALU, Design of ALU, Status Register, Design of Shifter

Module:6	Contemporary Issues	2 hours
	Total Lecture hours:	45 hours

Text Books	
1	Morris Mano, M., 2016. Digital Logic and Computer Design. Pearson Education India. ISBN: 9789332542525.
2	John F. Wakerly, 2008. Digital Design Principles and Practices, Fourth Edition, Pearson Education. ISBN: 978-8131713662.
Reference Books	
1.	Malvino, A.P. and Leach, D.P. and GoutamSaha. 2014. Digital Principles and Applications (SIE). Tata McGraw Hill. ISBN: 9789339203405.
2.	Charles H. Roth Jr. 2013, Fundamentals of Logic Design, seventh Edition, Cl- Engineering. ISBN: 978-1133628477

Mode of Evaluation: Quiz, Assignment, Design Project, Case Study, Seminar, CAT and FAT

List of Experiments (Indicative)

Realization of Logic gates using discrete components, verification of truth table for logic gates, realization of basic gates using NAND and NOR gates	**2 hours**
Implementation of Logic Circuits by verification of Boolean laws and verification of De Morgans.	**4 hours**
Adder and Subtractor circuit realization by implementation of Half-Adder and Full-Adder, and by implementation of Half-Subtractor and Full-Subtractor.	**6 hours**
Combinational circuit design **i.** Design of Decoder and Encoder **ii.** Design of Multiplexer and De multiplexer **iii.** Design of Magnitude Comparator **iv.** Design of Code Converter	**6 hours**
Sequential circuit design **i.** Design of Mealy and Moore circuit **ii.** Implementation of Shift registers **iii.** Design of 4-bit Counter **iv.** Design of Ring Counter.	**6 hours**
Implementation of different circuits to solve real world problems: **i.** A digitally controlled locker **ii.** A bank queuing system has a capacity of 5 customers	**6 hours**
Total Laboratory Hours	**30 hours**

Mode Evaluation: Continuous Assessment and Final Assessment Test.

Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C

IACSI203	**Computer Organization and Architecture**	3	1	0	4
Pre-requisite	IACSI202	**Syllabus Version**			
		1			

Course Objectives

1. To familiarize with architecture, performance metrics, and data representation to implement an arithmetic unit and instruction set
2. To discuss the importance of memory hierarchy, I/O devices, and interfacing techniques, along with their performance metrics
3. To explore high-performance architecture through pipeline, parallel processing, and GPGPU

Course Outcomes

1. Understand the structure and analyze performance assessments related to Computer design
2. Solve computer arithmetic and programs using number representation and instructions set, respectively.
3. Analyze various levels of memory and I/O systems
4. Gain an insight on various parallel processor organizations
5. Illustrate the working principles of CUDA and GPGPU architecture

Module:1	**Basic Concepts and Performance**	**4 hours**

Architectures Evolution, Stored Program Architecture and its functions-Registers-Performance metrics: Basic measurements, CPI, Execution time, MIPS & FLOPS

Module:2	**Computer Arithmetic and Machine Instruction**	**7 hours**

The Arithmetic &Logic unit - Integer representation: Integer arithmetic. Floating point representation: Floating point arithmetic. Machine instruction characteristics-Instruction type and Formats- Addressing modes-Assembly language

Module:3	**Memory and Input and Output Unit**	**13 hours**

Memory system overview- Organization of Cache memory - Elements of cache design; main memories, Secondary memories: Magnetic Memory, Optical Memory, Solid State Devices-RAID, Error detection and corrections; I/O devices: I/O Modules- Programmed I/O- Interrupt Driven I/O- Direct Memory Access		
Module:4	**Parallel Processing**	**11 hours**
Pipelining - Hazards - Parallelism - Multiple Processor Organizations: Symmetric Multiprocessors (SMP)-Cache coherence and MESI protocol. Multithreading and chip Multiprocessors (CMP) -Clusters & distributed system -Non-uniform memory access (NUMA)- Amdahl's Law, Little's Law, Benchmarks & Spec 91-Cloud computing: Parallelism in cloud environments, IaaS, PaaS, SaaS models		
Module:5	**General Purpose Graphics Processing Unit**	**8 hours**
Introduction to General Purpose GPUs: CPU vs GPU, when to use GPU as a coprocessor- CUDA basics-GPU Architecture overview-Intel's Gen 8 GPU: architecture features		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**
	Tutorial Hours:	**15 hours**
Text Book(s)		
1. William Stallings, **"Computer Organization and Architecture"**, Pearson Education Inc, 11th Edition, **2022**		
Reference Books		
1. David A Patterson and John L. Hennessy Morgan Kaufmann, **"Computer Organization and Design: The Hardware/Software Interface -RISC - V"**, Elsevier Science and Technology, 2nd Edition, **2020**		
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Seminar		
Recommended by Board of Studies :		16-10-2025
Approved by Academic Council : No. 80		20-11-2025

Course Code	Course Title	L	T	P	C
IACSI204	Database Systems Design	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To explore the basic concepts of Database Management Systems and Entity Relationship Modeling
2. To comprehend the concepts normalization, query optimization and relational algebra
3. To apply concurrency control, recovery, security and indexing for the existent domain problems

Course Outcomes

1. Acquire a good understanding of the architecture and functioning of database management systems and an ability to construct an ER model, derive the relational schemas from the model
2. Analyze and improve a database design by normalization and analyze the basics of query evaluation and heuristic query optimization techniques
3. Apply the concepts of Transaction Processing, concurrency control and recovery for the desirable database problem
4. Synthesis the basic database storage structure and access techniques including LSM structures
5. Explore the concepts of semi structured, unstructured and other emerging database technologies

Module:1	**Introduction to Database Management**	**7 hours**

Introduction to Databases - Database Environment - Database Architectures and the Web - Components of a DBMS - Oracle Architecture - Oracle's Logical Database Structure -Oracle's Physical Database Structure - Data Models and Conceptual Modelling.

Module:2	**EER Model and Database Design**	**10 hours**

Database System Development Lifecycle - The Relational Model - Entity-Relationship Modeling - Enhanced Entity-Relationship Modeling - Converting ER and EER diagram into tables. Normalization - Purpose of Normalization - Data Redundancy and Update Anomalies - Functional Dependencies - Process of Normalization- Advanced Normalization: More on Functional Dependencies - BCNF - 4NF - 5NF.

Module:3	**Query Processing and LSM Based Storage Techniques**	**11 hours**

Relational Algebra - Basic Concepts on Indexing - Ordered Index - Hash Indices - Multiple- Key Access - Write-Optimized Index Structures - Bitmap Indices -Log-Structured Merge(LSM) tree: Structure and functionality - Write and read paths: Compaction strategies - Case studies: LSM implementations in databases - Overview of Query Processing - Heuristic Approach to Query Optimization - Cost Estimation for the Relational Algebra Operations - Query Optimization in Oracle		
Module:4	**Transaction Management**	**7 hours**
Transaction Support: Properties of Transactions - DBMS Transaction Subsystem - Serializability and Recoverability - Concurrency Control: Locking Methods - Dead Lock -Database Recovery - Advanced Transaction Models		
Module:5	**Semi structured Databases and Data Warehousing Design**	**8 hours**
Semi structured Data- Introduction to XML - XML-Related Technologies - XML Schema - XML Query Languages - XML and Databases - XML in Oracle - NoSQL Introduction - Aggregate Data Models - Key-Value databases - Document databases - Column-family stores - Graph Databases - Data Warehousing Concepts: Introduction to Data Warehousing-Data Warehouse Architecture - Data Warehousing Tools and Technologies - Data Mart.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Thomas Connolly and Carolyn Begg, "**Database Systems: A Practical Approach to Design, Implementation, and Management**", Pearson Education Inc, 6th Edition, **2019**

Reference Books

1. A. Silberschatz, H. F. Korth and S. Sudershan, "**Database System Concepts**", McGraw Hill Education, 7th Edition, **2019**
2. R. Elmasri and S. B. Navathe, "**Fundamentals of Database Systems**", Pearson Education Inc, 7th Edition, **2020**
3. Pramod J. Sadalage and Martin Fowler, "**NoSQL A Brief Guide to the Emerging World of Polyglot Persistence**", Addison Wesley United States, 2nd Edition, **2019**

Indicative Experiments

1. The following scenario and related questions are *only illustrative*. Students may choose this or any other equivalent domain (Hospital, Library, Banking, Inventory, E-commerce, etc.) for the experiments listed below Sample Scenario (For Illustration): Smart University Management System (SUMS) A University wants to develop a	2 hours

Smart University Management System (SUMS) to manage its academic and	

administrative activities. The database must handle information related to students, courses, faculty, departments, exams, and fees. The system should ensure data integrity, efficient access, and user security. Entities & Relationships: • Student (Student_ID, Student_Name, Gender, DOB, Dept_ID, Email, Phone) • Department (Dept_ID, Dept_Name, HOD_ID, Location) • Faculty (Faculty_ID, Faculty_Name, Designation, Salary, Dept_ID) • Course (Course_ID, Course_Name, Credits, Dept_ID) • Enrollment (Enroll_ID, Student_ID, Course_ID, Semester, Grade) • Exam (Exam_ID, Course_ID, Exam_Date, Max_Marks, Min_Marks) • Fee (Fee_ID, Student_ID, Amount, Due_date, Status) Sample Datasets:1. University Database SQL 2020, GitHub Public Repository2. University-Database, 2023, GitHub Public RepositoryTranslate the ER-Model to relational tables and populate relational database by using :Data Definition Language(DDL)	
2. Translate the ER-model from the experiment one to equivalent relational tables and practice the following: Data Manipulation Language (DML), Transaction Control Language (TCL), Data Control Language (DCL).	2 hours
3. Convert an EER model to relational tables, apply the various constraints and populate the relational databases	4 hours
4. Apply the Normalization concept and normalize the unnormalized table into 1NF, 2NF and 3NF and apply the single row functions and group functions on the normalized tables	4 hours
5. Demonstrate the Arithmetic, Comparison, Logical and LIKE Operators	4 hours
6. Implement the Nested queries, Views and Joins	4 hours
7. Working with Procedures, Functions, Cursors and Triggers using PL/SQL	4 hours
8. Interface the database with Front Ends using ODBC and JDBC. Perform the operations and access the database from the Front End	4 hours
9. Implementation of database creation and deletion, document creation, insertion, updation and deletion and File attachment using NoSQL	2 hours

Total Laboratory Hours	**30 hours**

Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar

Recommended by Board of Studies :	16-10-2025

Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI205	Operating System Fundamentals	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To introduce operating system fundamentals and its various subsystems that manage relevant computer hardware
2. To discuss various issues related to process synchronization and memory management
3. To investigate different scheduling criteria and explore file management subsystem

Course Outcomes

1. Acquire knowledge on various components, functions and design issues of operating system
2. Understand process and thread types, their structures, states, multi-threading concepts and synchronization.
3. Analyse various memory management techniques, including partitioning, paging, segmentation, and page replacement policies
4. Apply various disk scheduling algorithms, along with the core principles of file management and input/output systems.
5. Design and analysis of RTOS and real time scheduling techniques, investigating the recent trends in OS

Module:1	**Introduction of Operating system**	**8 hours**

Operating system definition and functions, Evolution of OS, Components of OS, OS Services and System calls, Approaches to Modern OS, Multiprocessor and multicourse organization; OS design issues; Processes: process control block, process state transition diagram; Process switching and execution, Inter Process Communication, Threads vs. Processes,Types of Threads, Multi-Threading Models

Module:2	**Process Management**	**9 hours**

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF, Priority, Round Robin algorithm, Multiple Processor Scheduling, Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Monitors, Classical problems of Synchronization, Deadlock: NecessaryConditions for Deadlock, Prevention, Avoidance, Detection, Recovery

Module:3	Memory Management and Virtual Memory	9 hours
Contiguous memory allocation, Address translation technique: Paging, page table, two-level paging and address translation, operation of paging and translation look aside buffer, Segmentation; Segmentation		

with Paging, Virtual memory, Demand Paging, Copy on Write, Page replacement policies: FIFO, LRU, optimal, Allocation of frames, Thrashing		
Module:4	**Storage Systems**	**9 hours**
I/O Buffering, Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap-Space Management, File-System Interface, File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection, File- System Implementation, File-System Structure, Implementation, Directory, Allocation methods		
Module:5	**Real Time Operating System RTOS**	**8 hours**
Real time system, RTOS introduction, Real-time scheduling - Overview, Characteristics and RTOS requirements; Scheduling of real-time process; EDF, RMS and priority inversion; Case study: Android OS, Linux OS		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. A.Silberschatz, P. B. Galvin & G. Gagne, "**Operating system concepts**", John Wiley, 10th Edition, **2018**

Reference Books

1. William Stallings, "**Operating Systems: Internals and Design Principles**", Pearson Education Inc, 9th Edition, **2021**
2. Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, "**Operating Systems, Three Easy Pieces** ", Arpaci Dusseau Books Inc , 1st Edition, **2018**
3. K.C. Wang, "**Embedded and Real-Time Operating Systems**", Springer United States, 2nd Edition, **2023**

Indicative Experiments

1. Program to report on the behavior of the OS to get the CPU type and model, kernel version	2 hours
2. Program to get the amount of memory configured into the computer, amount of memory currently available	2 hours
3. Parallel Thread management using Pthreads library. Implement a data parallelism using multi-threading. Ex: An application should have a thread created with synchronization and thread termination. Every thread in the sub-program must return the value and must be synchronized with the main function. Final	2 hours
consolidation should be done by the main (main function).	
4. Create child, Orphan and Zombie processes using suitable system calls such as fork(), exec(), wait(), kill(), sleep() and exit() system calls.	2 hours
5. Implement the various process scheduling mechanisms such as FCFS, SJF, Priority, Round robin	4 hours
6. Implement the solution for reader - writer's problem using semaphore. Ex: A shared log file in a data archiving system is frequently read by multiple analysis tasks (Readers) but updated only occasionally by a dedicated maintenance script (Writer). The system must allow maximum concurrency for Readers while ensuring that Writers have exclusive access. Assume a Reader-Prioritypolicy, where a waiting Writer must wait if there is an active Reader or if a Reader is waiting to enter the critical section.	4 hours
7. Implement the solution for dining philosopher's problem using monitor. For example, five philosophers (P0 to P4) are seated at a round table, and a dedicated synchronization mechanismâ€”a Monitorâ€”is used to control their access to the five chopsticks. The Monitor ensures that no philosopher picks up chopsticks if either neighbor is currently eating, thus preventing deadlock.	4 hours
8. Implement banker's algorithm. Ex: Simulate the Banker's Algorithm to avoid deadlock. For a given set of processes, resource allocation, and maximum demand, determine whether the system is in a safe state and identify the safe execution sequence if it exists.	4 hours

9. Implement the first fit; best fit and worst fit file allocation strategy. Ex: A main memory uses fixed partitioning with four partitions of 100 KB, 350 KB, 150 KB, and 450 KB. Four processes arrive sequentially â€" P1 (140 KB), P2 (90 KB), P3 (300 KB), and P4 (80 KB). Allocate these processes using First Fit, Best Fit, Worst Fit, and Next Fit strategies. For each strategy, specify the assigned partition (or if it waits/rejects) and compute total internal fragmentation from successful allocations	4 hours
10. Implement the Page Replacement Algorithms FIFO, LRU and Optimal. Ex: Consider a page reference string with limited memory frames. Apply FIFO, LRU, and Optimal page replacement algorithms to determine page faults and frame contents for each reference. Compare their performance to analyze efficiency in minimizing page faults	2 hours
Total Laboratory Hours	**30 hours**

Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI206	Internet and Web Programming	2	0	4	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To provide foundational knowledge of web technologies such as HTML, CSS, and JavaScript for creating dynamic and interactive web pages
2. To develop skills in server-side programming using PHP and database integration for effective data handling.
3. To introduce modern front-end and full-stack development practices using JavaScript frameworks like ReactJS for building complete web applications

Course Outcomes

1. Develop structured and semantic web pages using HTML5 tags and elements.
2. Apply CSS styling techniques to design responsive and visually appealing web pages.
3. Design interactive web pages using JavaScript
4. Build database driven web applications using server-side scripts with PHP.
5. Develop and deploy single-page applications using ReactJS

Module:1	**Hypertext markup language**	**5 hours**

Internet and World Wide Web Resources, Web system architecture, Basics of HTML, HTML Tags, Structure, HTML Coding Conventions - Block Elements, Text Elements, Character References - Lists, Images, section, article, and aside, Elements - nav and a Elements - header and footer Elements. Forms and input validations

Module:2	**Cascading style sheets**	**6 hours**

CSS Overview - CSS Rules, CSS Syntax and Style - Class Selectors, ID Selectors, span, and div Elements - Box model, Text Effects, Positioning of element, Animations, Multiple column layouts. Flex box and CSS Grid

Module:3	**Java Script**	**5 hours**

JavaScript - Variables and Data Types - Statements - Operators - Literals - Functions - Objects - Arrays - Built-in Objects , Asynchronous JS - DOM manipulation - Exceptions, Event handling, Validation. ES6+ features.

Module:4	**Server side programming**	**6 hours**

Introduction to Web servers, PHP, String Processing and Regular Expressions, Form processing and Business logic, Dynamic content, File handling, Email, Database connectivity. MYSQL, Security - Introduction to REST APIs, Common vulnerabilities - XSS, SQL Injection

Module:5	**React JS**	**6 hours**

Node.js, NPM, React Environment Setup, ReactJS Basics, React JSX, React Component API, React Component Life Cycle, React Constructors, React Dev Tools, React Native vs ReactJS,

React Dataflow, React State , React Props, React Props Validation, Styling React, Hooks and Routing, Deploying React. JSON.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**30 hours**

Text Book(s)

1. Dean, J, "**Web Programming with HTML5, CSS, and JavaScript**", Jones and Bartlett Learning, 1st Edition, **2018**
2. Minnick C, "**Beginning ReactJS foundations building user interfaces with ReactJS: An Approachable Guide**", O Reilly Media Inc, 1st Edition, **2022**

Reference Books

1. Harvey M Deitel, Paul J Deitel and Tem R Nieto, "**Internet and World Wide Web How to Program**", Pearson Education Inc, 6th Edition, **2020**
2. Rebah, H.B., Boukthir, H. and Chedebois, A, "**Website Design and Development with HTML5 and CSS3**", John Wiley and Sons, 1st Edition, **2022**

Indicative Experiments	
1. A web designer wants to create the homepage for VIT's student portal. Help him design a webpage using basic HTML elements such as headings, paragraphs, hyperlinks, and forms to display student information	4 hours
2. A university portal needs a page showing course details, faculty list, and department highlights. Demonstrate how you would use Lists, Tables, Images, Section, Article, and Aside elements to structure and present this information effectively.	4 hours
3. You are asked to enhance the appearance of the student registration page. Investigate various CSS components like selectors, properties, and values to style forms, buttons, and layout consistently	4 hours
4. Design a multi-section webpage for a college event that combines HTML structure with CSS styling for banners, color themes, and responsive design.	4 hours
5. Create a dynamic feedback form for employees that validates inputs and displays confirmation messages using JavaScript event handling and DOM manipulation.	4 hours
6. Develop a hotel feedback form where students can rate multiple subjects using Radio Buttons and Checkboxes. Use loops (for, while, do-while) and External JavaScript files to process and display the collected data.	4 hours
7. Build an interactive image gallery that allows users to reorder or highlight images using CSS z-index, Canvas drawing, List Boxes, and Pull-Down Menus, with JavaScript event handlers and listeners.	4 hours
8. Create a voter information management system where the administrator can add, update, or delete student records stored in a MySQL database using PHP and phpMyAdmin	4 hours

9. Develop a faculty management app that uses React constructors and demonstrates the React component lifecycle. Test and debug the app using React Developer Tools, and explain the difference between React Native and ReactJS.	8 hours
10. Design a course registration system using ReactJS that handles state management and props validation, applies styling, and implements React Hooks and Routing for navigation between pages.	4 hours
11. Deploy a fully functional event management web application built with ReactJS to a hosting platform. Present it as a case study demonstrating real-world deployment steps.	8 hours
12. Develop a RESTful API using Node.js or PHP that interacts with a front-end ReactJS application to fetch, display, and update student data in real time.	8 hours
Total Laboratory Hours:	**60 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI302	Data Communication and Computer Networks	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To Provide knowledge of fundamental concepts in data communication, networking protocols, architectures, and applications.
2. To Gain expertise in designing, implementing, and analyzing the performance perspective of TCP/IP layered Architecture.
3. To Deal with the major issues of the layers of the model.

Course Outcomes

1. Understand the layered structure of a typical networked architecture and different types of network topologies
2. Analyse data transmissions such as impairments, media, encoding methods, transmission modes, signals, multiplexing, and modulation techniques
3. Apply error detection/correction, flow control concepts in Virtual LANs and Wireless LANs (802.11)
4. Design sub-netting and enhance the performance of routing mechanisms.
5. Explore the suitable transport and application layer protocols for different communications.

Module:1	**Basics of Data Communication and Computer Network**	**7 hours**

Definition and Uses of Computer Network, Criteria for a Data Communication Network, Components of Data Communication, Classification of Computer network, Network Topology, Network Models: OSI, TCP/IP.

Module:2	**Physical Layer**	**9 hours**

Networking Devices: Hubs, Bridges, Switches, Routers, and Gateways Performance Metrics. Transmission Impairments, Transmission Medium, Data Encoding: Line Encoding, Types of Line Coding, Analog-to-Digital Conversion- Pulse code modulation (PCM), Transmission Modes- Half and Full Duplex- Signals Bandwidth and Data Rate Multiplexing Shift Keying.

Module:3	**Data link Layer**	**9 hours**

Switching - Circuit Switching, Packet Switching and Message Switching, Error Detection and Correction- One- and two-dimensional parity checks, Hamming code, Cyclic redundancy check (CRC); Flow Control: Protocols: Protocols for Noiseless Channels and Noisy Channels Ethernet- Access Control Protocols: CSMA, CSMA/CA, CSMA/CD, Token Ring, TDMA, FDMA, CDMA-Virtual LAN-Wireless LAN (802.11).

Module:4	**Network Layer**	**9 hours**

IP Addressing Scheme, Subnet Addressing, Subnet Masks, IPV4 Addressing, IPV6 Addressing, Internetworking -Need for Network Layer, Internet as a Datagram Network, Internet as a Connectionless Network, IPv4 Header format, IPv6: Advantages, Packet Format, Extension Headers Transition from IPv4 TO IPv6: Dual Stack, Tunneling, Header Translation, Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP). Routing Algorithms: Distance Vector Routing Protocol, Link State Routing Protocol.		
Module:5	**Transport and Application Layer**	**9 hours**
Services of Transport Layer, Transport Layer Protocols: TCP, UDP. Congestion Control Algorithms- Congestion prevention policies; Quality of Service- Traffic shaping, Leaky bucket algorithm, Token bucket algorithm, Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), TELNET, SNMP, DNS, Hypertext Transfer Protocol (HTTP), World Wide Web (WWW) - Basics of Network Security- Introduction to cryptography - DES, RSA and Diffie-Hellman Algorithms		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Behrouz A. Forouzan, "**Data Communications, and Networking**", McGraw Hill Education, 6th Edition, **2022**

Reference Books

1. Alberto Leon-Garcia, Indra Widjaja, "**Communication Networks Fundamental Concepts Architectures**", McGraw Hill Education, 2nd Edition, **2019**
2. William Stallings, "**Data and Computer Communications**", Pearson Education Inc, 11th Edition, **2017**

Indicative Experiments

1. Simulate a network administrator's task of configuring and troubleshooting systems using Linux commands to test connectivity, routing, and interface status.	2 hours
2. Act as a network engineer ensuring reliable data transfer by implementing and testing CRC, checksum, and Hamming code under simulated noisy conditions. Use Error Detection and Correction Mechanisms	2 hours
3. Model sender-receiver communication where buffer limitations exist and apply Stop-and-Wait or Sliding Window techniques to prevent data loss with effective implementation of Flow Control Mechanisms.	2 hours
4. Design an efficient IP addressing plan for an expanding organization using subnetting and super netting to minimize address wastage using IP addressing and Classless Addressing. Also verify connectivity and troubleshoot potential issues in a simulated lab environment.	4 hours
5. Configure routers with protocols like RIP, OSPF, and EIGRP to observe routing table formation and analyze their performance across different topologies. Implement and troubleshoot these routing protocols in a simulated	4 hours

environment (e.g., Cisco Packet Tracer or GNS3) to ensure seamless connectivity and optimal routing.	
6. Develop a client-server system using TCP and UDP sockets to simulate real-time communication such as file transfer or chat applications.	4 hours
7. Secure client-server communication through TLS handshake and certificate exchange to prevent data interception and ensure confidentiality.	4 hours
8. Simulate a congested network and apply TCP congestion control algorithms to maintain stable data flow and reduce packet loss.	4 hours
9. Explore tools like Packet Tracer, NS2, and GNS3 to design, simulate, and analyze complex network architectures for performance evaluation.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI303	IoT and Embedded Systems	3	0	2	4
Pre-requisite	Microprocessors and Microcontrollers	**Syllabus Version**			
		1			

Course Objectives

1. To provide an understanding of IoT and embedded systems with emphasis on hardware and firmware design.
2. To impart knowledge and skills in microcontroller programming with GPIO interfacing, enabling students to design and implement real-time applications using sensors, actuators and standard communication protocols
3. To examine critical issues in IoT systems, including security, privacy, data management, resource constraints, and scalability.

Course Outcomes

1. Understand and design the architecture and key components of IoT and embedded systems.
2. Apply programming techniques for various microcontrollers in GPIO and peripheral interfacing.
3. Develop IoT solutions for real-time applications using communication protocols, sensor and actuators
4. Integrate IoT systems with cloud and fog platforms for data storage, analytics and dashboards.
5. Explore the contemporary issues and emerging technologies in IoT and embedded systems.

Module:1	**Introduction to IoT and Embedded Systems**	**7 hours**
Introduction to IoT, Physical and Logical Design of IoT, IoT Levels, IoT Enabling Technologies, Overview of Embedded Systems, Design Challenges, Processors vs Microcontrollers		
Module:2	**Architecture and Peripheral Interfacing**	**8 hours**
Architecture - 8051, PIC, and ARM, GPIO, Memory, Sensors & actuators interfacing, ADC/DAC, Interrupts, Power management.		
Module:3	**Communications and Networking of IoT**	**10 hours**
Four Pillars of IoT Paradigm, Control Units, Communication modules, Bluetooth, ZigBee, WiFi, GPS, LoRa, IoT protocols, IPv6, 6LoWPAN, RPL, MQTT, CoAP.		
Module:4	**IOT Data and Security**	**9 hours**
Data Acquisition & Storage, Cloud Platform Integration, Fog Computing, Stream Processing & Dashboards, Real-time analytics and visualization tools for monitoring IoT systems, Security Mechanisms, Open Authorization, Operational Scenarios.		

Module:5	Embedded Programming and RTOS	9 hours
MCU, Timers/counters, PWM, UART, A/D, D/A, Interrupts, Inter Integrated Circuits (I2C), RTOS basics, Preemptive Task Scheduling Policies, Priority Inversion, Inter process communication, POSIX, Protothreads.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Arshdeep Bahga, Vijay Madisetti, "**Internet of Things: A Hands-on Approach**", Universities Press, 1st Edition, **2024**
2. Rajkamal, "**Embedded Systems: Architecture, Programming and Design**", McGraw Hill Education, 4th Edition, **2020**

Reference Books

1. Dimitrios Serpanos & Marilyn Wolf, "**Internet of Things (IOT) System Architectures, Algorithms, Methodologies**", Springer International publishing AG, 1st Edition, **2018**
2. Shibu K V, "**Introduction to Embedded Systems**", McGraw Hill Education, 2nd Edition, **2017**
3. Simone Cirani, "**Internet of Things: Architectures, Protocols and Standards**", Wiley, 1st Edition, **2019**

Indicative Experiments

1. Study on various sensors, actuators and its applications.	4 hours
2. Implement basic input/output operations and signal generation techniques.	2 hours
3. Detect the distance of an object using an Ultrasonic Sensor and control movement with a Servo Motor.	2 hours
4. Program to read the temperature and humidity values and display values in	2 hours
an LCD.	
5. Explore interrupt-driven programming and control of Stepper Motor.	2 hours
6. Control multiple LEDs remotely using a web interface or mobile app over the internet.	2 hours
7. Create a temperature monitoring system that logs data to the cloud and provides real-time analytics.	2 hours
8. Demonstrate techniques for efficient data collection and broadcasting in an IoT network.	2 hours
9. Retrieve a sensor readings from multiple connected nodes and display the data on the IoT gateway.	2 hours
10. Implement the COAP protocol to interact with sensor data.	2 hours
11. Acquire sensor data from distributed IoT nodes through a gateway device and securely transmit to store the data on a cloud platform.	4 hours

12. To monitor sensor data from IoT nodes and trigger real-time alerts when the data crosses predefined thresholds, using a gateway and cloud-based notification system.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI304	Cryptography and Cyber Security	3	0	2	4
Pre-requisite	Data Communication and Computer Networks	**Syllabus Version**			
		1			

Course Objectives

1. To comprehend the foundational principles of cryptography, including its core objectives of ensuring confidentiality, integrity, and authentication.
2. To develop proficiency in key cryptographic applications, such as symmetric and asymmetric encryption algorithms, secure key exchange protocols, and the use of digital signatures.
3. To acquire knowledge on various techniques and tools used in penetration testing

Course Outcomes

1. Understanding of the fundamental concepts of cryptography
2. Analyse the symmetric and asymmetric key cryptography techniques.
3. Apply cryptographic concepts to real-world scenarios.
4. Experiment the emerging cybersecurity principles in the respective adversarial risks
5. Explore emerging security solutions using automated tools and techniques

Module:1	**Introduction to Cryptography**	**5 hours**

Overview of cryptography: Basic concepts, historical perspective, and applications. Network security fundamentals: Threats, vulnerabilities, and countermeasures. Mathematics of cryptography: modular arithmetic, Euclidean, extended Euclidean algorithm, Chinese remainder theorem. Monolithic ciphers: addition and multiplication ciphers, Polyalphabetic ciphers: Vigenere's ciphers, Hill ciphers

Module:2	**Symmetric and Asymmetric Key Cryptography**	**10 hours**

Groups, Rings, Fields, GF, Inverse of a number and polynomial using extended Euclidean algorithm. Introduction to Stream cipher, Block cipher. Data Encryption Standard, Advanced Encryption Standard: Euler's Phi-Function, Fermat's Little Theorem, Euler's theorem, Miller-Rabin theorem. RSA, ElGamal Cryptosystem, Elliptic Curve cryptography

Module:3	**Key Exchange and Message Authentication**	**10 hours**

Key Distribution: Diffie-Hellman, Elliptic Curve Diffie-Hellman, Diffie-Hellman Hash functions, SHA512, Message Authentication Code, HMAC, Digital Signature: Digital Signature Standard, Elliptic Curve Digital Signature Algorithm, ElGamal based Digital signature

Module:4	**Foundational Cybersecurity and Hacking Methodologies**	**10 hours**

Types of Hackers - Ethical Hacking -Five phases of Hacking - Vulnerabilities - Vulnerability Databases - Scanning Systems and Network - DNS Enumerations - Active and Passive Sniffing - ARP Poisoning - Password cracking techniques and tools - Social Engineering Attacks - Penetration Testing - Metasploit: Auxiliary modules - payloads - Meterpreter sessions		
Module:5	**Applied Network and Application Security**	**8 hours**
Types of Wireless Encryption - WIFI password cracking - Evil Twin attack - Introduction to OWASP top 10 attacks - SQL Injection - Cross-Site Scripting attack - Cryptographic failures - Broken Authentication - Automated Web Application Vulnerability Assessment tools - Database vulnerabilities -Penetration testing MySQL databases using Metasploit - Counter measures and tools		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. William Stallings, "**Cryptography and Network Security: Principles and Standards**", Prentice Hall India, 7th Edition, **2017**
2. Najera-Gutierrez G, Ansari JA, "**Web Penetration Testing with Kali Linux: Explore the methods and tools of ethical hacking with Kali Linux**", Packt Publishing Ltd, 3rd Edition, **2018**

Reference Books

1. John R. Vacca, "**Computer and Information Security Handbook**", Morgan Kaufmann Publishers, 3rd Edition, **2017**
2. Kevin Beaver , "**CISSP, Hacking for Dummies**", John Wiley and Sons Inc, 7th Edition, **2022**

Indicative Experiments

1. An undercover agent uses the VigenÃ¨re cipher with the keyword â€œVICTORIOUSâ€ to encrypt the secret message â€œMeet at the old bridge at midnight.â€ This cipher shifts each letter based on the keyword, making the message unreadable without it. Two functionsâ€”one for encryption and one for decryptionâ€”ensure secure communication.	2 hours
2. As a junior cryptographer, you must build key RSA tools: Euclid's algorithm for GCD, Extended Euclidean for modular inverse, and Chinese Remainder Theorem for solving congruences. These functions form the backbone of secure encryption, enabling key generation, decryption, and efficient number	4 hours
reconstruction in public-key cryptographic systems.	
3. Encrypt a plaintext file of mock 1990s financial data using the Triple DES algorithm and a 64-bit key. Implement both encryption and decryption to ensure the original data is perfectly recoverable. This secures the file for long-term storage, complying with data protection standards of that era.	2 hours

4. Alice encrypts a confidential proposal using Bob's RSA public key. Only Bob can decrypt it with his private key. Implement RSA to generate Bob's key pair, encrypt the message with his public key, and decrypt it using his private key to confirm secure, private communication over the public internet.	4 hours
5. Two servers securely establish a shared symmetric key using the Diffie-Hellman protocol. They agree on public parameters (p, g), generate private keys, compute public keys, and exchange them. Each server then calculates the shared secret independently. The resulting keys match, enabling encrypted communication over an insecure network without prior trust.	2 hours
6. Analyze SHA-512's message schedule to observe how 16 initial 64-bit words expand, driving its avalanche effect. Implement the SHA-512-word generation function to expand these 16 words into the full 80 words used in the compression function.	2 hours
7. As a junior analyst, use Nmap to audit a test network. Identify live hosts via ping scan, then scan each for open TCP/UDP ports. Perform service and version detection to map running software. Compile a clear report detailing host activity, port status, and service information for your manager.	2 hours
8. As an ethical hacker, use John the Ripper with rockyou.txt to crack test server password hashes. Then, use Hydra for a dictionary attack on a test login portal or SSH server. Report cracked passwords, time taken, and recommend stronger password policies to improve organizational security and prevent weak credentials.	4 hours
9. Use Wireshark to capture HTTP traffic between the user and server. Filter for POST requests showing cleartext credentials. Check server responses for a Set-Cookie header. If missing or malformed, the session may not persist, causing repeated logins. This helps diagnose cookie handling issues in non-encrypted web portals.	4 hours
10. Conduct a penetration test on a cloned server using Nmap for reconnaissance. Identify vulnerable services, configure a Metasploit exploit, and gain remote shell access. Perform post-exploitation by listing user files to confirm control. Document findings and recommend fixes before the server goes live to ensure secure deployment.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI305	Artificial Intelligence and Machine Learning	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			

		1

Course Objectives

1. To introduce the fundamental principles of Artificial Intelligence and Machine Learning, emphasizing problem-solving, search strategies, and intelligent agent design.
2. To develop the ability to represent and reason with knowledge, both deterministically and under uncertainty, using logical and probabilistic models.
3. To apply supervised, unsupervised, and ensemble learning techniques for analyzing data, building predictive models, and understanding contemporary advancements in AI and ML.

Course Outcomes

1. Apply the principles of Artificial Intelligence to design intelligent agents and solve problems using uninformed, informed, and adversarial search strategies.
2. Apply knowledge using propositional and predicate logic and perform inference using logical and probabilistic reasoning techniques under uncertainty.
3. Analyse the core concepts of Machine Learning, data preprocessing, and model evaluation to prepare data for effective learning.
4. Implement supervised learning algorithms for regression and classification tasks and evaluate their performance using appropriate metrics.
5. Solve real world applications using unsupervised and ensemble learning techniques.

Module:1	**Foundations of Artificial Intelligence and Search Algorithms**	**10 hours**

Introduction to AI , Types of AI, Turing Test, Classification of AI with respect to Environments, PEAS, Classification of Agents. State space tree/graph, Uninformed Search - BFS, DFS, Iterative Deepening, Heuristic search: Best first, A*, Local Search: Simulated Annealing, Genetic Algorithm. Adversarial Search: minimax, alpha-beta pruning.

Module:2	**Knowledge Representation and Reasoning under Uncertainty**	**10 hours**

Representation by Propositional Logic, Inference - Forward Chaining (FC), Backward Chaining (BC), Modus Ponens (MP), Representation by First Order Predicate Logic, Inference - Unification, FC, BC, MP, Horn Clause, Resolution by Refutation, Uncertainty Representation, Conditional Independence, Bayes' Rule, Bayesian Belief Network, Inference by Bayesian Belief Networks.

Module:3	**Introduction to Machine Learning and Data Preparation**	**9 hours**

Machine Learning, Types of Learning - Supervised, Unsupervised, Reinforcement. Components of a Learning System, Data Representation and Feature Selection - Data Preprocessing -Normalization, Encoding, Handling Missing Values, Train-Validation-Test Split, Cross-Validation, Overfitting and Underfitting, Bias-Variance Trade-off, Model Evaluation -

Accuracy, Precision, Recall, F1-Score, ROC Curve.

Module:4	**Supervised Learning**	**8 hours**

Linear Regression - Logistic Regression - k-Nearest Neighbour (k-NN) Classification, NaÃ¯ve Bayes Classifier, Decision Tree Classifier, Support Vector Machines (SVM) - Regularization, Hyperparameter Tuning, Model Comparison and Performance Evaluation.		
Module:5	**Unsupervised Reinforcement and Ensemble Learning**	**6 hours**
Clustering - K-Means, Hierarchical Clustering - Reinforcement Learning - Deterministic and Non-Deterministic Rewards and Actions - Ensemble Methods - Bagging, Boosting - AdaBoost, Random Forest.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Russell, S. and Norvig, P. , "**Artificial Intelligence - A Modern Approach**", Prentice Hall, 4th Edition, **2021**
2. Ethem Alpaydin, "**Introduction to Machine Learning**", MIT Press Prentice Hall , 4th Edition, **2020**

Reference Books

1. Poole, D. and Mackworth, A., "**Artificial Intelligence: Foundations of Computational Agents**", Cambridge University Press, 2nd Edition, **2017**
2. AurÃ©lien GÃ©ron, "**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**", OReilly Media, 3rd Edition, **2023**

Indicative Experiments

1. You are given two jugs of capacity 4 liters and 3 liters. The objective is to measure exactly 2 liters of water using these jugs. a. Implement a solution to the water jug problem using Breadth-First Search (BFS) and Depth-First Search (DFS). b. Trace the sequence of states generated by both methods. c. Compare the number of steps required in BFS and DFS to reach the goal state.	2 hours
2. A sliding puzzle (3x3 board) must be solved by reaching the goal state from a given initial state. Solve the puzzle using A* search with use of Manhattan distance and misplaced tiles as heuristics. Compare the number ofnodes expanded / steps taken and overall performance. Input Example: Initial Goal 1 2 4 1 2 3 3 0 6 4 5 6 7 5 8 7 8 0 Output: Sequence of moves to reach goal, with cost.	2 hours
3. Consider a two-player board game (e.g., Tic-Tac-Toe or Chess). Use the	2 hours
Minimax algorithm to simulate how an AI agent selects the best move, assuming the opponent also plays optimally. Then apply Alpha-Beta Pruning to the same game tree and demonstrate how it reduces the number of nodes explored while still reaching the optimal decision	

4. In a university knowledge base, facts like Professor(x), Teaches(x, y), and Enrolled(z, y) are given along with rules such as If a professor teaches a course and a student is enrolled in that course, then the professor teaches that student. Using Forward Chaining, infer all possible professor-student teaching relationships. Using Backward Chaining, check whether a specific professor teaches a given student.	2 hours
5. A dataset contains customer information with both numerical and categorical attributes. Some records have missing entries, and the scales of the numerical features vary significantly. Perform data preprocessing to handle missing values, encode categorical variables, and normalize the numerical features.	4 hours
6. A retail chain wants to recommend similar products to new customers. Apply the K-Nearest Neighbours (KNN) algorithm to customer purchase history data to identify the top three similar products and visualize the resulting clusters.	2 hours
7. A housing firm monitors property prices across different cities. Develop a Linear Regression model to predict house prices using square footage, location score, and amenities index, and validate the model's performance.	4 hours
8. A bank wants to classify loan applicants as eligible or ineligible based on income, credit score, and past repayment history. Build an SVM classifier to separate the two classes and tune kernel parameters to improve model accuracy.	4 hours
9. A marketing firm wants to segment its customers based on spending patterns, purchase frequency, and visit duration. Implement K-Means Clustering to identify distinct customer groups and interpret the cluster centroids to design targeted promotional strategies.	2 hours
10. An insurance company wants to predict claim approval based on applicant age, policy type, claim amount, and past claim history. Build a Random Forest classifier to improve prediction accuracy, analyze feature importance, and compare the performance with a single Decision Tree model.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Group Discussion, Presentation, Case Study	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI306	Software Engineering Principles and Practices	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To provide insight into software process models and project management techniques to plan, execute, and manage development projects efficiently.
2. To disseminate knowledge for designing robust software systems with proficiency in tools, testing, maintenance, and quality assurance.
3. To inculcate professional software engineering skills for analyzing, designing, and managing systems using best practices and modern tools.

Course Outcomes

1. Understand and demonstrate the software process models and project management techniques to effectively plan, execute, and manage software development projects.
2. Analyze and design software systems using appropriate methodologies and tools to produce robust, requirement specifications for development.
3. Apply testing strategies and maintenance plans to ensure the performance of software systems.
4. Experiment software quality assurance practices to deliver high-quality products and address challenges in software engineering.
5. Solve real-world software problems using suitable project management models, modern tools, and design techniques to address engineering issues.

Module:1	Basics and Software Process Models	7 hours

Software Engineering Vs System Engineering, Software process, methods and project, classical software life cycle, traditional and evolutionary process models. Advanced Process models: Agile principles, Extreme programming, Kanban, key values, Scrum process framework, Scrum roles, events of Scrum and its benefits, DevOps principles and architecture, DevOps life cycle and its benefits.

Module:2	Project Management and Software Analysis	11 hours

Planning, Scope, Work break-down structure, PERT and CPM, Critical Path Calculation, Gantt chart, resource allocation, milestones, deliverables, risk assessment and cost estimation with function points and COCOMO. Software requirements: Functional, non-functional, Detailed Requirements Engineering process, goal-oriented analysis, use case model, use case specifications, activity diagram: system level, use case level, sequence model, SRS template and its customization.

Module:3	Software Design	7 hours

Fundamental design concepts and principles, DFD L0, L1, L2, goal-oriented design, object-oriented design techniques, system modularity, cohesion and coupling techniques, class model, component and package diagram, refactoring of design, user-interface design, software architecture, cloud architecture.

Module:4	Software Implementation and Testing	8 hours
Structured coding Techniques, Coding Styles, IDE Tools, client-server side implementation platforms and use, Seven Steps of implementing, Testing: Plan, Test Design, Test case and Execution, Reviews, Inspection and Auditing - Black- box testing, White box testing, testing the software change, Regression Testing - Mutation Testing, advanced testing with tools and Case Studies on Testing.		
Module:5	**Software Maintenance and Quality**	**10 hours**
Continuous Deployment, Automated Monitoring, Software Maintenance: Types of Maintenance, Structured versus unstructured maintenance, Typical problems, Maintenance process - Software Configuration Management - Component Reusability - Overview of re-engineering & Reverse Engineering. Software engineer culture and ethics, product and process quality, Software quality standards: ISO, CMM, TQM, methods of quality management		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Roger Pressman and Bruce Maxim, "**Software Engineering: A Practitioner's Approach**", McGraw Hill Inc, 9th Edition, **2020**

Reference Books

1. Ian Somerville, "**Software Engineering**", Addison Wesley, 10th Edition, **2019**
2. Buede, Dennis M., and William D. Miller, "**The Engineering Design of Systems: Models and Methods**", John Wiley and Sons, 4th Edition, **2024**

Indicative Experiments

1. Form teams, explore real-world problems, and develop problem statements with individual focus areas. Identify and justify the appropriate process models.	2 hours
2. Scheduling a Project with Cost Estimation, PERT and Gantt charts constructed on the identified problem statement using Project Management Tools. Develop a Work Breakdown Structure (WBS) for a project, estimate the total time and cost required to complete it without delays, schedule the individual activities to meet the project completion time efficiently, and identify the critical bottleneck activities where delays may occur to prevent project completion delay	4 hours
3. Introduction to JIRA tool - Project management activities using JIRA.	4 hours
4. Based on the identified problem statement, conduct a requirement analysis to identify and document the functional and non-functional requirements of the system and identify the stakeholders. Develop structural and object- oriented models based on the gathered requirements using CASE Tool.	4 hours
5. UML Behavioural Diagrams and SRS Document using CASE Tool. For the chosen problem statement, construct UML behavioural diagrams such as DFD, activity diagrams and sequence diagrams, along with use case specifications, to develop a Software Requirements Specification (SRS) document.	4 hours

6. Create design diagrams including Entity-Relationship (ER) diagrams, class diagrams, client-server architecture diagrams, and custom design diagrams to develop a Software Design Document (SDD) with CASE Tool.	4 hours
7. Based on the identified problem statement, design the user interface (UI), create test cases and a test plan, and develop a delivery plan using latest tools.	8 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar, Course Based Design Project, Course Project, Case Study	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI308	Formal Languages and Automata Theory	3	1	0	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To introduce the fundamental concepts of formal languages, grammars, and automata theory to provide a foundation for the theoretical understanding of computation.
2. To explore the design and analysis of finite automata, regular expressions, context-free grammar, pushdown automata, Turing machines, and undecidability, emphasize their role in language recognition.
3. To introduce the role of formal languages and automata theory in compiler design.

Course Outcomes

1. Apply various proof techniques in the context of formal languages and automata
2. Design and analyze finite automata and regular expressions to solve problems and recognize regular languages.
3. Construct Pushdown automata and understand context-free languages.
4. Recognize the concept of Turing machine and recursive & recursively enumerable languages.
5. Explore the application of formal languages and automata theory in compiler design.

Module:1	**Introduction to Formal Languages and Proof Techniques**	**4 hours**

Proof Techniques: Induction, Contradiction, Contrapositive - Introduction to Formal Languages and Grammars - Types of Grammars - Alphabets - Strings - Operations on Languages -

Overview of Models of Automata - Chomsky Hierarchy

Module:2	**Finite Automata and Regular Languages**	**11 hours**

Regular Expressions and Regular Languages - Deterministic and Non-Deterministic Finite Automata (DFA & NFA) - Îµ-moves in NFA - NFA to DFA Conversion - Equivalence of Finite Automata and Regular Expressions - Regular Grammar and Finite Automata - - Minimization of DFA - Closure and Decision Properties of Regular Languages - Pumping Lemma for Regular Languages.

Module:3	**Pushdown Automata and Context Free Languages**	**11 hours**

Context-Free Grammar (CFG) and Language (CFL) - Derivations - Parse Trees - Ambiguous and Unambiguous Grammars - Left Recursion and Left Factoring - Simplification of CFG: Elimination of Useless Symbols, Unit Productions, Null Productions - Chomsky Normal Form - Greibach Normal Form - Pumping Lemma for CFL - CYK Algorithm - Closure Properties of CFL. Deterministic and Non-Deterministic PDA - Instantaneous Description of PDA - Power of Deterministic and Non-Deterministic PDA

Module:4	Turing Machine and Undecidability	7 hours
Turing Machines as acceptor and transducer - Multi head and Multi tape Turing Machines - Universal Turing Machine - The Halting problem - Recursive and Recursively Enumerable Languages, Language that is not Recursively Enumerable (RE) - Computable functions - Undecidable problems - Post's Correspondence Problem.		
Module:5	**Applications of Formal Languages in Compilers**	**10 hours**
Structure and Phases of a Compiler - Lexical Analysis: Patterns, Lexemes, Tokens and Symbol Table - Lexical Analysis using Finite Automata and Regular Expression - Program Constructs using Regular Expression - Syntax Analysis - Parse Trees, Top-Down Parsing (Predictive Parsing, LL(1)), Bottom-Up Parsing (Shift-Reduce, Operator Precedence) - Syntax-Directed Translation, Three Address Code Generation.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**
	Tutorial Hours:	**15 hours**

Text Book(s)

1. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, "**Introduction to Automata Theory, Languages, and Computation**", Pearson Education Inc, 3rd Edition, **2020**
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Sorav Bansal, "**Compilers: Principles, Techniques, and Tools**", Pearson Education Inc, 2nd Edition, **2023**

Reference Books

1. Peter Linz, Susan H. Rodger, Jones, "**An Introduction to Formal Languages and Automata**", Bartlett Publishers Inc, 7th Edition, **2022**
2. Wladyslaw Homenda, Witold Pedrycz, "**Automata Theory and Formal Languages**", Gruyter Publisher, 1st Edition, **2022**
3. Dan A. Simovici, "**Introduction to the Theory of Formal Languages**", World Scientific Publisher, 1st Edition, **2024**

Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Seminar

Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI401	Cloud Computing Technologies	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To gain knowledge on various cloud service and deployment models.
2. To understand the concept of cloud virtualization, cloud management, storage and security concepts.
3. To explore real-time application deployment using cloud APIs.

Course Outcomes

1. Understand the basics of cloud computing, cloud service and deployment models.
2. Analyze various virtualizations techniques, cloud management and monitoring tools.
3. Apply cloud computing, storage and network services for real time applications.
4. Experiment with suitable tools and techniques to assure security in cloud environment.
5. Design and develop real time applications in cloud domains.

Module:1	Overview of Cloud Computing	6 hours

Evolution of Computing Paradigm-Benefits and Challenges of Cloud Computing-NIST Cloud Computing Reference Architecture-Types of Cloud Deployment Models: Private, Public, Hybrid- Cloud Service Models: Infrastructure as a Service (IaaS)-Platform as a Service (PaaS), Software as a Service (SaaS).

Module:2	Virtualization Cloud Management and Monitoring	9 hours

Basics of Virtualization- Hypervisors- Benefits of Virtualization- Types of Virtualizations: Desktop Virtualization, Application Virtualization, Server Virtualization-Storage Virtualization-Network Virtualization. Need for Monitoring: Log Monitoring, Metric Monitoring-AWS CloudWatch, Cloud Resource Management: Allocation-Policies and Mechanisms.

Module:3	Cloud Compute Storage and Networking services	13 hours

Compute Services**:** Virtual Machines, Instance families, Auto scaling groups-Bare metal, virtual and serverless computing -AWS EC2; Networking Services: Virtual Private Cloud (VPC)-Internet gateways-NAT-Security groups- NACLs- Firewall rules-Load balancers- Content Delivery Networks (CDN)-Peering-VPN- Direct Connect. Storage Services: Block storage (EBS)-File storage (EFS)-Backup and snapshot strategies- Object store design (S3), Microservices: Container orchestration-Docker, Kubernetes.

Module:4	Cloud Security	8 hours

Requirement for Cloud Security- Shared responsibility model-Service Level Agreements-Privacy-Confidentiality-Integrity-Availability- Identity and Access Management- Intrusion detection system- Compliance frameworks: ISO/IEC 27018, PCI DSS, HIPAA-Security Challenges-Cloud Security best practices.		
Module:5	**Cloud Application Development**	**7 hours**
AWS Elastic BeanStalk-Microsoft Azure App Service-Google App Engine-OpenStack- Cloud Foundry.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Erl, Thomas, Ricardo Puttini, and Zaigham Mahmood, "**Cloud computing: concepts, technology & architecture**", Pearson Education Inc, 2nd Edition, **2023**
2. Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi, "**Mastering Cloud Computing**", McGraw Hill Education, 2nd Edition, **2024**

Reference Books

1. Sehgal, Naresh, Bhatt, Pramod Chandra P., Acken, John M, "**Cloud Computing with Security Concepts and Practices**", Springer International Publishing, 2nd Edition, **2020**
2. Buyya, Rajkumar, James Broberg, and Andrzej Goscinski, "**Cloud Computing: Principles and Paradigms**", Wiley India, 1st Edition, **2011**
3. Bahga, Arshdeep, and Vijay Madisetti, "**Cloud computing: A hands-on approach**", CreateSpace Independent Publishing Platform, 1st Edition, **2013**

Indicative Experiments

1. Launch a virtual machine on AWS/ GCP/Azure.	2 hours
2. Configure a Virtual Private Cloud (VPC) with public and private subnets.	2 hours
3. Deploy a Scalable Web Application on Cloud with compute and Networking Services.	4 hours
4. Implement storage services (AWS S3 / Azure Blob) and perform CRUD operations.	4 hours
5. Monitor resource utilization using CloudWatch / Azure Monitor	2 hours
6. Configure CDN to cache static website content.	2 hours
7. Build a serverless API (e.g., AWS Lambda/GCP Cloud Functions) with an event trigger.	2 hours
8. Setup autoscaling and load balancing for fault tolerance.	2 hours
9. Build and deploy a containerized microservice application using an orchestration tool like Kubernetes or docker swarm.	2 hours
10. Implement IAM roles, security groups, and NACLs for secure access control.	2 hours

11. Setup OpenStack private cloud.	2 hours
12. Deploy a simpleweb app to a managed PaaS (e.g., Azure App Service, GCP App Engine, or AWS Elastic Beanstalk).	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI401	Programming for Data Science	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To apply data manipulation techniques and analyze practical problems using a programming-based approach.
2. To create and interpret reports by visualizing results through the use of appropriate programming tools.
3. To develop and apply R programming skills for solving data science problems.

Course Outcomes

1. Design and implement solutions to problems using the R programming language.
2. Construct and organize data collection forms for analyzing real-time data effectively.
3. Analyze and interpret data to derive meaningful insights using statistical inference techniques.
4. Evaluate and interpret results and examine model performance using appropriate visualization and analytical methods.
5. Develop, implement and share applications using R to solve practical problems.

Module:1	**Functions Vectors and Lists**	**9 hours**

Programming with Râ€"Running R Codeâ€"Including Comments - Defining Variables, Functions -Built-in R Functions - Loading Functions - Writing Functions - Using ConditionalStatements - Vector - Vectorized Operations - Vector Indices - Vector Filtering - Modifying Vectors, Lists - Creating Lists - Accessing List Elements - Modifying Lists - Applying Functions to Lists with lapply().

Module:2	**Data Wrangling and Data Manipulation**	**10 hours**

Understanding Dataâ€"The Data Generation Processâ€"Finding Dataâ€"Types of Dataâ€"Interpreting Dataâ€"Using Data to Answer Questionsâ€"Data Framesâ€"Working with DataFramesâ€"Working with CSV Dataâ€"Data Manipulationâ€"Core dplyr Functions - Performing Sequential Operations - Analyzing Data Frames by Group - Joining Data Frames Together - dplyr in Action: Analyzing Flight Dataâ€"Reshaping Data with tidyr - gather() - spread() - tidyr in Action: Exploring Educational Statistics.

Module:3	**Accessing Databases and Web APIs**	**6 hours**

An Overview of Relational Databasesâ€"A Taste of SQLâ€"Accessing a Database from Râ€"Accessing Web APIsâ€"RESTful Requests -Accessing Web APIs from R -Processing JSON Data APIs in Action: Finding Cuban Food in Seattle.

Module:4	Simple and Interactive Data Visualization	10 hours
Designing Data Visualizationsâ€”The Purpose of Visualization - Selecting Visual Layouts - Choosing Effective Graphical Encodings - Expressive Data Displaysâ€”Enhancing Aesthetics - Creating Visualizations with ggplot2â€”A Grammar of Graphics - Basic Plotting with ggplot2 - Complex Layouts and Customization - Building Mapsâ€”ggplot2 in Action: A case study - The Plotly Package - The Rbokeh Package - The Leaflet Package - Interactive Visualization in Action: Exploring Changes to the City of Seattle.		
Module:5	**Building and Sharing Applications**	**8 hours**
Dynamic Reports with R Markdown - Setting Up a Report - Integrating Markdown and R Code - Rendering Data and Visualizations in Reports - Sharing Reports as Websites - R Markdown in Action: Reporting on Life Expectancy - Building Interactive Web Applications with Shiny - The Shiny Framework - Designing User Interfaces - Developing Application Servers - Publishing Shiny Apps - Shiny in Action: Visualizing Fatal Police Shootings.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Michael Freeman and Joel Ross, "**Programming Skills for Data Science: Start Writing Code to Wrangle, Analyze and Visualize Data with R**", Addison Wesley, 1st Edition, **2018**

Reference Books

1. Benjamin S. Baumer, Daniel T. Kaplan and Nicholas J. Horton, , "**Modern Data Science with R**", CRC, 1st Edition, **2021**
2. John Mount and Nina Zumel, "**Practical Data Science with R**", Wiley, 2nd Edition, **2019**

Indicative Experiments

1. Write and execute R functions to perform arithmetic and statistical operations. Demonstrate the use of arguments, return values, and default parameters.	4 hours
2. Create and manipulate vectors and lists in R. Perform element-wise operations, indexing, and concatenation.	2 hours
3. Create data frames using built-in and user-defined datasets. Demonstrate operations such as adding/removing columns, filtering rows, and summarizing data.	4 hours
4. Write an R script to identify, remove, or impute missing values in a dataset using functions	4 hours
5. Use the dplyr and tidyr packages to perform data manipulation tasks such as filtering, grouping, summarizing, joining, and reshaping datasets.	2 hours
6. Parse and extract information from a JSON file using the jsonlite package. Convert JSON data into R data frames for further analysis.	2 hours

7. Access and retrieve data from a public API using R packages such as httr or jsonlite. Demonstrate how to send requests, handle responses, and process the returned data.	3 hours
8. Use ggplot2 to create various types of visualizations (bar chart, histogram, boxplot, scatter plot). Customize plots with titles, labels, legends, and themes.	3 hours
9. Develop interactive plots using packages like plotly or shiny. Demonstrate dynamic data exploration through zooming, tooltips, or filtering features.	3 hours
10. Case study-based experiments	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI402	Data Visualization	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand and explain the fundamental concepts of data visualization and apply principles of effective visual design
2. To develop proficiency in utilizing programming libraries, platforms and dashboard tools to design and implement interactive, scalable visualizations.
3. To select and apply suitable visualization techniques and dashboard design principles to analyze complex datasets and facilitate business intelligence.

Course Outcomes

1. Understand the principles of data visualization, including human perception, cognition, and guidelines for creating effective visualizations.
2. Apply appropriate visualization techniques to visualize univariate, multivariate, temporal, spatial, hierarchical, and network data.
3. Evaluate the usability, accessibility and ethical aspects of visualizations through real-world case studies.
4. Develop effective visualizations to synthesize complex data and communicate insights clearly.
5. Design, and optimize dashboards that support operational, analytical, strategic, and tactical decision-making

Module:1	**Introduction to Data Visualization**	**8 hours**

Data visualizationâ€”Data abstractionâ€”Task abstractionâ€”Dimensions and measures - Human perception, Cognition, and visual encoding - Principles of good visualizationâ€”Tufte's guidelines, Gestalt Principles- Four levels of validation.

Module:2	**Data types**	**10 hours**

Visualization of categorical, numerical, temporal, spatial dataâ€”scalar and point techniques - color maps -Contouring Height Plots - Vector visualization techniques - vector properties - Vector Glyphs- Vector color coding stream objects.

Module:3	**Techniques for Data Visualization**	**8 hours**

Univariate and multivariate data typesâ€”univariate and multivariate data visualization - Graphs, charts, and plots: bar, line, scatter, boxplot, histogram, heatmap - Multidimensional visualization: parallel coordinates, scatterplot matrix, tree maps, star plots - Case studies.

Module:4	**Visual Analytics**	**8 hours**

Visual Variables - Text data visualization - Networks and Trees visualization - Time series and geospatial data visualization - Map color and other channels - Manipulate View - Arrange Tables Geospatial Data Reduce items and attributes.		
Module:5	**Visualization tools and Design of Dashboard**	**9 hours**
Dashboardâ€”Dashboard Taxonomiesâ€”operational, analytical, strategic, and tacticalâ€—User Interaction and Organizational Functionsâ€—Dashboard Design Principles. Worksheets - Workbooks - Workbook Optimization - Protection - Case studies and use cases for creation of dashboards.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Nathan Yau , **"Visualize This: The FlowingData Guide to Design, Visualization, and Statistics"**, Wiley, 2nd Edition, **2024**
2. Tamara Munzner, **" Visualization Analysis and Design"**, A K Peters Visualization Series CRC Press, 1st Edition, **2014**
3. Scott Murray, **"Interactive Data Visualization for the Web"**, OReilly Media, 2nd Edition, **2017**

Reference Books

1. Brett Powell , **"Mastering Power BI: Expert Techniques for Effective Data Analytics and Business Intelligence"**, Packt Publishing, 1st Edition, **2018**
2. Cole Nussbaumer Knaflic, **"Storytelling with Data: A Data Visualization Guide for Business Professionals"**, Wiley, 1st Edition, **2015**
3. Claus O. Wilke , **"Fundamentals of Data Visualization"**, OReilly Media, 1st Edition, **2019**
4. Andy Kirk, **"Data Visualization: A Handbook for Data-Driven Design"**, Sage Publications, 4th Edition, **2025**

Indicative Experiments

1. A company wants to compare its quarterly profits across five years. Develop code for the following. (Any language like python, R) Use different charts to show profits for each year per quarter	4 hours
• Add a legend, different colors for each year, and value labels above bars • Interpret which quarter shows consistent growth across years	
2. A financial analyst wants to compare investment returns across different asset types, such as stocks, bonds, real estate and crypto. Implement the following tasks using Tableau, Power BI and R. • Use different plots to show the distribution of returns • Compare average and variability across asset classes • Highlight which asset type is most volatile	4 hours

3. Collect the data set of customer feedback for an online food delivery service. Use visualization tools to identify the most frequently occurring words and key themes in feedback. • Preprocess the text (remove stop words, punctuation and duplicates). • Create a word cloud showing the most common words. • Develop a highlight table that displays each word and its corresponding frequency.	2 hours
4. Analyze the sales performance of petroleum products across various countries globally. Using visualization tools, perform map visualization to analyze the geographical distribution of sales and identify top-performing regions.	2 hours
5. Visualization of multidimensional data: You have a dataset of social media posts with features such as post length, hashtags, likes, comments, shares and platform. • Build an interactive dashboard using Use Plotly or Tableau • Include a bubble chart (Likes vs. Shares, bubble size = Comments, color = Platform) • Filter by hashtags or platform type • Create a data story explaining what drives high engagement	2 hours
6. Identifying outliers and anomalies in a dataset: A data center monitors server load and response times.Attributes: Timestamp, CPU_Usage, Memory_Usage, Response_Time, Server_ID. • Plot real-time time series data for CPU and memory usage • Apply Z-score and rolling mean deviation to detect spikes in load • Mark anomalous periods on the chart • Discuss possible causes: hardware fault, overload or attack	2 hours
7. Collect the US data store dataset containing sales details by product category and subcategory. Using visualization tools, • Create a tree map visualization to represent the contribution of each subcategory to total sales within its category. • Create a tree map showing the category â†' subcategory hierarchy.	4 hours
• Use sales as the measure to determine size and color intensity. • Add labels for subcategory names and sales values. • Analyze which subcategories contribute most to total sales.	
8. Creation of dashboards using Tableau/Power BI (sales, finance, IPL, healthcare, cybercrime, etc.). Recommended to use any publicly available free datasets.	4 hours
9. Project work: End-to-end visualization of a large dataset with a dashboard. Recommended to use any publicly available free datasets.	6 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	

Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI403	Data Mining	3	0	2	4
Pre-requisite	IACSI204	**Syllabus Version**			
		1			

Course Objectives

1. To understand and apply fundamental concepts of data warehousing and data mining techniques for knowledge discovery and practical applications.
2. To apply data preprocessing, dimensionality reduction and pattern discovery techniques for effective data analysis.
3. To apply classification, clustering and advanced mining techniques on varied data for real-world problem solving.

Course Outcomes

1. Understand the concepts and architecture of data mining and data warehousing.
2. Apply data preprocessing and dimensionality reduction techniques for quality data analysis.
3. Perform association rule mining and pattern discovery using Apriori and FP-Growth algorithms.
4. Implement and evaluate classification and clustering techniques for predictive analysis.
5. Analyze and extract insights from stream, spatial, multimedia, and web data.

Module:1	Fundamentals of Data Mining and Knowledge Discovery	9 hours

Data Warehouse: Data Warehouse Architecture, Data Lake, Datawarehouse Modeling: Schema and Measures, OLAP Operations, Data Cube Computations; Data Mining: Steps in Data mining process, Data Mining Functionalities, Architecture of a Typical Data Mining Systems, Classification of Data Mining Systems, Knowledge Discovery in Databases (KDD), KDD Process, Datamining Applications.

Module:2	Data Measurement and Pre processing	8 hours

Data Types, Statistics of data, Similarity and distance measures, Data Quality, Data cleaning and Data integration, Data Transformation, Dimensionality Reduction.

Module:3	Association Rule Mining and Pattern Discovery	8 hours

Association Rule Mining and Pattern Discovery - Frequent Itemset Generation; The Apriori Principle; Apriori Algorithm - rule generation, candidate generation and pruning, support counting, computational complexity, and confidence-based pruning; Compact representation of frequent itemsets, maximal and closed frequent itemsets; Alternative methods for generating frequent itemsets; FP-Growth Algorithm and FP-Tree representation; Evaluation of association patterns. Mining High Dimensional Data, Mining rare and negative patterns.

Module:4	Classification and Clustering	9 hours

Classification and prediction, Decision Tree Induction-Bayesian classification, Lazy Learners, Linear Classifiers, Model Evaluation and Selection. Cluster analysis, Partition based methods, Hierarchical based methods, Density based methods, Cluster Evaluation.		
Module:5	**Mining of Stream Data and Complex Data Mining**	**9 hours**
Mining Streams, Time Series and Sequence Data: Mining Data Streams, Mining Time-Series Data, Mining Sequence Patterns in Transactional Databases, Mining Sequence Patterns in Biological Data, Graph Mining, Social Network Analysis and Multi-relational Data Mining. Mining Object, Spatial, Multimedia, Text and Web Data: Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Spatial Data Mining, Multimedia Data Mining, Text Mining, Mining the World Wide Web.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Jiawei Han, Micheline Kamber and Jian Pei, "**Data Mining Concepts and Techniques**", Morgan Kaufmann, 4th Edition, **2023**
2. Pang-Ning Tan; Michael Steinbach; Vipin Kumar, "**Introduction to Data Mining**", Pearson Education, 2nd Edition, **2019**
3. Adhikari Ratnadip, R. K. Agrawal, "**An Introductory Study on Time Series Modeling and Forecasting**", LAP Lambert Academic Publishing, 1st Edition, **2013**

Reference Books

1. GarcÃa, Salvador, JuliÃ¡n Luengo, and Francisco Herrera, "**Data preprocessing in data mining**", Cham Switzerland Springer International Publishing, 1st Edition, **2015**
2. K.P. Soman, Shyam Diwakar and V. Ajay, "**Insight into Data mining Theory and Practice**", Easter Economy Edition Prentice Hall of India, 1st Edition, **2006**
3. G. K. Gupta, "**Introduction to Data Mining with Case Studies**", Prentice Hall of India, 3rd Edition, **2006**

Indicative Experiments

1. Creating data frames, loading and visualization of dataset	2 hours
2. Design and implement a data cube for a retail store to analyze sales by product, region, and time. Recommended to use any publicly available free datasets.	2 hours
3. Perform data cleaning, handle missing values, and normalize features from housing data for model readiness. Recommended to use any publicly available free datasets.	4 hours
4. Apply Principal Component Analysis (PCA) to reduce features in a health dataset and visualize results. Recommended to use any publicly available free datasets.	2 hours
5. Generate frequent itemsets and association rules to identify commonly bought products together. Recommended to use any publicly available free datasets.	2 hours

6. Build and evaluate a decision tree to classify students as pass/fail based on their performance indicators.	4 hours
7. Perform clustering on customer demographics and spending patterns to identify distinct customer groups. Recommended to use any publicly available free datasets.	4 hours
8. Classify emails as spam or non-spam using the NaÃ¯ve Bayes algorithm and evaluate performance metrics. Recommended to use any publicly available free datasets.	4 hours
9. Apply time-series analysis to forecast future airline passenger traffic. Recommended to use any publicly available free datasets.	2 hours
10. Perform text preprocessing and sentiment classification on movie reviews using a supervised learning model. Recommended to use any publicly available free datasets.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI404	Soft Computing Techniques	3	1	0	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand and apply soft computing concepts and techniques to design suitable solutions for real-world problems.
2. To understand and explain the fundamentals of artificial neural networks, fuzzy sets and fuzzy logic.
3. To understand and apply optimization techniques, genetic algorithms, rough sets and hybrid models.

Course Outcomes

1. Apply soft computing concepts for enhanced decision-making using approximate reasoning and adaptive learning.
2. Understand the fundamental concepts of soft computing, including fuzzy logic, neural networks and genetic algorithms.
3. Implement fuzzy logic and reasoning to manage uncertainty in real-world problems.
4. Apply optimization and hybrid model techniques to solve engineering problems effectively.
5. Analyze and utilize Rough Set Theory and Genetic Algorithms for handling uncertainty in real-world problems.

Module:1	**Introduction to Soft Computing**	**10 hours**

Introduction to Soft computing- Soft computing Vs Hard Computing- Characteristics of Soft computing-Components of Soft computing-Fuzzy computing-Neural networks, Evolutionary computing-Machine Learning-other techniques of Soft computing, Fuzzy sets-Features of Fuzzy membership functions-Fuzzy set operations- Fuzzy set properties-Binary Fuzzy Relation-Fuzzy membership functions-Methods of membership value assignments-Fuzzification Vs Defuzzification-Fuzzy Inference Rules-Fuzzy - C Means.

Module:2	**Optimization techniques**	**14 hours**

Artificial Neural Networks - Radial basis function network - Neural Nets based on competition - Max net-Mexican Hat-Hamming net- Kohonen Self-organizing Feature Map - Learning Vector Quantization - Ant colony Optimization - Particle Swarm Optimization - Simulated Annealing

Module:3	**Genetic Algorithms**	**7 hours**

Introduction to Genetic Algorithms-Population Initialization-Fitness function calculation-Parent Selection-Crossover-Mutation-How Genetic algorithm works- Application areas of Genetic algorithms

Module:4	**Rough Set Theory**	**6 hours**
The Pawlak Rough Set Model-Basic terms in Pawlak Rough set Model-Measures of Rough set Approximations-Using Rough set Information System-Decision rules and Decision Tables-Parameters of Decision Tables-Application areas of Rough set theory		
Module:5	**Hybrid Systems**	**6 hours**
Introduction to hybrid systems-Neurogenetic Systems-GA based weight determination of Multilayer Feedforward net-Neuroevolution of Augmenting Topologies (NEAT)-Fuzzy Neural Systems-Fuzzy Neurons-Adaptive Neuro fuzzy Inference System (ANFIS),Fuzzy Genetic Systems		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**
	Tutorial Hours:	**15 hours**

Text Book(s)

1. Nandi, G, "**Principles of Soft Computing Using Python Programming: Learn How to Deploy Soft Computing Models in Real World Applications**", Wiley Publications, 1st Edition, **2023**
2. Timothy J. Ross, "**Fuzzy Logic with Engineering Applications**", Wiley Publications, 4th Edition, **2021**

Reference Books

1. D. K. Pratihar, "**Soft Computing : Fundamentals and Applications**", Narosa Publishing House , 1st Edition, **2025**
2. S.N.Sivanandam , S.N.Deepa, "**Principles of Soft Computing**", Wiley Publications, 3rd Edition, **2018**
3. S. Rajasekaran and G.A. Vijayalakshmi Pai, "**Neural networks fuzzy systems and evolutionary algorithms: synthesis and application**", PHI Publication, 2nd Edition, **2017**

Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test

Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Professional Elective Courses

Course Code	Course Title	L	T	P	C
IAMDI405	Machine Learning for Data Science	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand the fundamental concepts of machine learning and its role in data science.
2. To apply supervised learning algorithms to analyze structured datasets and make predictions.
3. To Implement unsupervised learning techniques for clustering, association and pattern discovery.

Course Outcomes

1. Apply supervised learning algorithms to perform predictive analysis on structured datasets.
2. Implement unsupervised learning techniques to identify patterns, clusters and associations in data.
3. Evaluate machine learning models using performance metrics, cross-validation and model selection techniques.
4. Develop and deploy advanced machine learning solutions, including ensemble methods and dimensionality reduction for real-world problems.
5. Analyze and interpret results to generate actionable insights and support data-driven decision-making.

Module:1	**Introduction to Machine Learning**	**7 hours**

Machine Learning Types; Data handling; Model: Selection, Training and Fine tuning a Model; Main Challenges: Data Inadequacy, Non-representativeness, Irrelevant features, Overfitting the Model, Underfitting the Model.

Module:2	**Supervised Learning Techniques I**	**9 hours**

Binary Classifier; Multiclass and Multi-label classification; Linear Regression; Gradient Descent: Batch Gradient, Stochastic Gradient Descent, Mini-batch Gradient Descent; Polynomial Regression; Logistic Regression: Estimating Probabilities, Decision Boundaries, SoftMax Regression; Performance Measures: Cross-Validation, Confusion Matrix, Accuracy, Precision, Recall and F1 score, AUC-ROC.

Module:3	**Supervised Learning Technique II and Ensemble Learning**	**9 hours**

SVM: Linear SVM with Soft Margin Classification, Non-linear SVM Classification, SVM Regression; NaÃ¯ve Bayes and KNN Classifier; Decision Tree: ID3, CART Algorithm - Gini impurity; Bagging: Random Forests; Boosting: Adaboost and Gradient Boosting; Stacking.

Module:4	Unsupervised Learning Technique	10 hours
Clustering (Partitioned, Hierarchical and Density based) basics; Partitioned: K means & K mode; Hierarchical: Agglomerative and Divisive; Density based: DBSCAN; Gaussian Mixtures; PCA (Principal Component Analysis); Kernel PCA.		
Module:5	**Neural Networks**	**8 hours**
Introduction to a Simple Neural Network; Key Computations; Output Layer of a Binary and a Multiclass problem; Choosing the right configuration; Loss Functions; Back Propagation.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Ethem Alpaydin, "**Introduction to Machine Learning**", MIT Press USA, 4th Edition, **2020**
2. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, "**Foundations of Machine Learning**", MIT Press USA, 2nd Edition, **2018**

Reference Books

1. U Dinesh Kumar, Manaranjan Pradhan, "**Machine Learning Using Python**", Wiley, 1st Edition, **2019**
2. Francois Chollet, "**Deep Learning with Python**", Manning Publications, 2nd Edition, **2021**

Indicative Experiments

1. Explore the various Machine Learning libraries in python.	2 hours
2. Data exploration and preprocessing in machine learning. You are hired as a junior data analyst at a hospital to improve patient outcomes using an anonymized dataset containing demographics (age, gender, location), clinical values (blood pressure, cholesterol, glucose), and outcomes (readmission within 30 days). The data includes missing values, unrealistic ages as outliers, and inconsistent categorical labels. Perform the following steps: load the dataset and display statistics, identify and handle missing values using two techniques, detect outliers in clinical values with visualization, standardize categorical labels, normalize numerical columns, and prepare the cleaned dataset for machine learning modeling.	4 hours
3. Implement Linear and non-linear regression algorithms and evaluate the algorithms using various performance measures A company collects data about employees' years of experience and their annual salary (in â‚¹). The HR department wants to predict the salary of a new employee based on experience using Linear regression algorithms and evaluate model performance using R^2, MSE, and RMSE. Recommended to use any publicly available free datasets. A hospital tracks the progression of a chronic disease based on patient age, glucose level, and time since diagnosis. The relationship between time and severity is non-linear. Apply non-linear regression models (e.g., polynomial or exponential) and Compare performance using R^2, RMSE, and residual plots. Recommended to use any publicly available free datasets.	4 hours

4. Implement Logistic regression and analyze its performance A teacher wants to predict whether a student will pass (1) or fail (0) based on their study hours and attendance percentage using Logistic regression. Use the algorithm to predict if a student with 85% attendance and 6 study hours will pass. Build a confusion matrix and find Accuracy, Sensitivity (Recall), Specificity. Plot the ROC & AUC curve and discuss the model's performance. Recommended to use any publicly available free datasets.	4 hours
5. Classify the sample data set using Linear and Non-linear SVM classifiers A tech company wants to build a spam filter to classify incoming emails as Spam (+1) or Not Spam (-1). The dataset includes features extracted from email content such as: Word frequency (e.g., â€œfreeâ€ , â€œwinâ€ , â€œofferâ€), Presence of links, Number of capital letters, Email length, Sender reputation score. The company wants a fast, interpretable model that performs well on linearly separable data. Use a Linear Support Vector Machine (SVM) to classify emails and evaluate its performance. Recommended to use any publicly available free datasets. A hospital provides a dataset with tumor measurements (radius, texture, perimeter, area). The data is not linearly separable. Implement an SVM with RBF kernel & polynomial kernel and compare both the model. Evaluate the model using Evaluate using F1-score, ROC-AUC, and precision-recall curve. Recommended to use any publicly available free datasets.	2 hours
6. Use supervised learning and evaluate the dataset for the performance metrics: Naive Bayes and KNN classifier You are working as a data scientist for an agricultural research center, analyzing a dataset of 30 crop records. Each record contains the attributes: Crop ID, Rainfall, Temperature, Humidity, Season, and Crop Yield. Your goal is to build a K-Nearest Neighbors (KNN) and Naive Bayes classifier to categorize the crop yield into three distinct classes (e.g., low, medium, high). Download the dataset given in the link based on realistic crop data then use 70% of the data for training the KNN model and the remaining 30% for testing. Finally, evaluate the model's performance by generating and interpreting the confusion matrix from the test results. Recommended to use any publicly available free datasets.	2 hours
7. To demonstrate the working of decision tree based algorithms: ID3, CART You are tasked with analyzing road safety data collected from road transport records, where each record includes attributes such as Road ID, Length, Number of Bends, Traffic Volume, and Accident Risk (classified into two categories: high risk and low risk). Download the dataset given in the link related to road safety data. Build a decision tree ID3 & CART classifier using 70% of the data as training samples to predict Accident Risk. Then, test the classifier's accuracy on the remaining 30% of the data. Finally, evaluate the model's performance by displaying and interpreting the confusion matrix based on the test results. Recommended to use any publicly available free datasets.	2 hours
8. Implement bagging using Random Forest A financial institution wants to predict whether a loan applicant is likely to default. The dataset includes features such as: Annual Income, Credit Score, Loan Amount, Employment Length, Debt-to-Income Ratio, Previous	2 hours

Defaults,Loan Purpose. The data is noisy and contains both numerical and categorical features. The institution wants a robust model that can handle feature interactions and avoid overfitting. Use the Random Forest algorithm, which applies bagging (Bootstrap Aggregation), to build a classifier that predicts loan default risk. Plot the confusion matrix and evaluate using Accuracy, Precision, Recall, F1-score, and ROC-AUC. Recommended to use any publicly available free datasets.	
9. Implement Principal Component Analysis for Dimensionality Reduction. A retail company collects data on customer purchases across 100+ product categories. They want to segment customers based on buying behavior. Apply PCA to reduce dimensionality before clustering. How does PCA help improve clustering performance? Recommended to use any publicly available free datasets.	2 hours
10. Implement K-means, K-Modes Clustering to find natural patterns in data An online retailer wants to segment customers based on their shopping behavior. The dataset includes: Annual spending, Number of purchases, Time spent on site and Discount usage. Apply K-Means clustering to identify customer segments. Recommended to use any publicly available free datasets. An HR department wants to group employees based on categorical features such as Department (e.g., Sales, IT, HR),Job Role (e.g., Manager, Analyst), Education Level and Work Shift. Use K-Modes clustering to find natural groupings. Explain how does K-Modes differ from K-Means in handling categorical data. Recommended to use any publicly available free datasets.	2 hours
11. Implement Hierarchical clustering You are a data analyst at a healthcare company. You have been given a dataset containing health metrics of patients such as age, blood pressure, cholesterol level, BMI, and glucose level. Your task is to identify groups of patients with similar health profiles to tailor personalized health programs. Compare the clusters obtained from hierarchical clustering with clusters obtained using K-means on the same dataset. Discuss any differences or similarities. Recommended to use any publicly available free datasets.	2 hours
12. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets. You are part of a healthcare AI research team working on automating the early detection of Diabetes. Your task is to build a Neural Network model to predict whether a patient is diabetic based on clinical data. You will use the Pima Indians Diabetes Dataset for training and testing your model. Recommended to use any publicly available free datasets.	2 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI406	Distributed Systems	3	1	0	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand the fundamental concepts, architectures and principles of distributed systems.
2. To analyze synchronization, consistency and coordination challenges in distributed environments.
3. To apply techniques for fault tolerance, replication and concurrency control in distributed systems.

Course Outcomes

1. Analyze communication mechanisms such as message passing, remote procedure calls and client-server models.
2. Apply synchronization, consistency and coordination techniques in distributed environments.
3. Implement fault tolerance, replication and concurrency control strategies in distributed systems.
4. Design and develop distributed applications for solving real-world problems.
5. Evaluate the performance, reliability and scalability of distributed systems.

Module:1	**Introduction to Distributed System**	**9 hours**

Introduction to Distributed Systems: Examples -Trends in Distributed Systems - Focus on resource sharing - System Models - Networking and Internetworking - Inter process Communications - Publish - subscribe system - message queues - shared memory approach; Remote procedure call: Distributed objects-communication between distributed objects - RMI - JSON-RMI.

Module:2	**Message Ordering and Snapshots**	**9 hours**

Message ordering and group communication: Message ordering paradigms -Asynchronous execution with synchronous communication - Synchronous program order on an asynchronous system -Group communication - Causal order (CO) - Total order; Global state and snapshot recording algorithms: Introduction - System model and definitions - Snapshot algorithms for FIFO channels.

Module:3	**Deadlock and Concurrency Control**	**9 hours**

Distributed mutual exclusion algorithms: Introduction - Preliminaries - Lamport's algorithm - Ricart; Agrawala algorithm Deadlock detection in distributed systems: Introduction - System model - Preliminaries - Models of deadlocks - Knapps classification - Algorithms for the single resource model - Distributed deadlock - Resource allocation model - requirements and performance metrics - classification of distributed deadlock detection algorithm.

Module:4	**Peer To Peer and Distributed Shared Memory**	**9 hours**

Peer-to-peer computing and overlay graphs: Introduction - Data indexing and overlays - Chord - Content addressable networks - Pastry, Tapestry, Gnutella; Distributed shared memory: Abstraction and advantages - Memory consistency models -Shared memory Mutual Exclusion.		
Module:5	**Process and Resource Management**	**7 hours**
Process Management: Process Migration- Features, Mechanism; Threads: Models, Issues, Implementation; Resource Management: Introduction- Features of Scheduling Algorithms - Task Assignment Approach - Load Balancing Approach - Load Sharing Approach.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**
	Tutorial Hours:	**15 hours**

Text Book(s)

1. M. Van Steen and A.S. Tanenbaum, "**Distributed Systems**", Maarten van Steen Distributed systems net, 4th Edition, **2023**
2. Pradeep K. Sinha, "**Distributed Operating Systems: Concepts & Design**", PHI, 1st Edition, **1998**

Reference Books

1. Tanenbaum A.S., Van Steen M, "**Distributed Systems: Principles and Paradigms**", Pearson Education, 3rd Edition, **2017**
2. George Coulouris, Jean Dollimore and Tim Kindberg, "**Distributed Systems Concepts and Design**", Pearson Education, 5th Edition, **2012**
3. Mukesh Singhal and N. G. Shivaratri, "**Advanced Concepts in Operating Systems, Distributed, Database, and Multiprocessor Operating Systems**", McGraw Hill, 1st Edition, **2008**

Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI407	Exploratory Data Analysis	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand different data types, measurement scales, and visualization techniques for effectively summarizing and interpreting datasets.
2. To explore skills in data transformation and statistical analysis for handling missing values, detecting outliers, and uncovering patterns in data.
3. To interpret dimensionality reduction, time series analysis, and advanced visualization tools to generate insights and dashboards for real-world applications.

Course Outcomes

1. Explain the concepts, significance, and steps of Exploratory Data Analysis (EDA) and differentiate between classical and Bayesian analytical approaches.
2. Apply data transformation techniques such as de-duplication, handling missing values, discretization, and outlier detection to prepare datasets for analysis.
3. Analyze datasets using descriptive statistical methods and summarize data through measures of central tendency, dispersion, and grouping operations.
4. Implement dimensionality reduction and time series analysis techniques to extract key patterns and trends from high-dimensional and temporal data.
5. Analyze advanced visualization tools and techniques to generate automated reports, visualize complex data relationships, and derive insights from structured and unstructured data.

Module:1	**Introduction to Exploratory Data Analysis**	**5 hours**

Introduction to Exploratory Data Analysis (EDA) - The significance of EDA Steps - Making sense of data: Numerical Data -Discrete data, Continuous data - Categorical data - Measurement Scales: Nominal, Ordinal, Interval, Ratio - Comparing EDA with classical and Bayesian Analysis

- Software Tools for EDA - Visual Aids for EDA (Line chart, Bar charts, Scatter plot, Histogram).

Module:2	**Data Transformation**	**9 hours**

Data transformation - Data analysis - Merging databases - Style data frames - Transformation Techniques: Performing data deduplication - Replacing values - Handling missing values - Renaming axis indices - Discretization and binning - Outlier detection and filtering - Permutation and random sampling - Computing indicators/dummy variables - String manipulation.

Module:3	**Descriptive Statistical Analysis**	**9 hours**

Understanding statistics: Distribution function - Cumulative distribution function - Measures of central tendency - Measures of dispersion - Grouping Datasets: Group by mechanics - Data aggregation - Pivot tables and cross-tabulations - Univariate analysis - Bivariate analysis - Multivariate analysis.		
Module:4	**Dimensionality Reduction and Time Series Analysis**	**10 hours**
Linear Methods: Principal Component Analysis - Singular Value Decomposition Factor Analysis - Intrinsic Dimensionality - Non-Linear Methods: Multidimensional Scaling - Self-Organizing Maps - Time Series Analysis (TSA): Fundamentals of TSA - Characteristics of TSA - Time based indexing - Visualizing time series - Grouping time series data - Resampling time series data - Applications of TSA (Open Power System Data).		
Module:5	**Advanced Data Visualization Techniques**	**10 hours**
Introduction to tools like Pandas Profiling - Sweetviz - and D-Tale - Generating automated reports and quick insights - Advanced Visualization Techniques: Sankey diagrams - Violin plots - Radial plots - Parallel coordinates plot - Choropleth map - Gauge chart - Network graphs for data relationships - EDA for Unstructured Data: Text Data - Image Data - Case studies on Sales Performance and Customer Insights Dashboard for a Retail Company.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Suresh Kumar Mukhiya and Usman Ahmed, "**Hands-On Exploratory Data Analysis with Python**", Packt Publishing Limited, 1st Edition, **2020**
2. Wendy L. Martinez, Angel R. Martinez, Jeffrey Solka, "**Exploratory Data Analysis with MATLAB**", CRC Press A Chapman and Hall Book, 3rd Edition, **2022**

Reference Books

1. Craig K. Enders, "**Applied Missing Data Analysis**", Guilford Press, 2nd Edition, **2022**
2. Jake Vanderplas, "**Python Data Science Handbook: Essential Tools for Working With Data**", OReilly Media, 2nd Edition, **2022**
3. Harish Garg, "**Mastering Exploratory Analysis with Pandas**", Packt Publishing Limited, 1st Edition, **2018**
4. Allen B. Downey, "**Think Stats: Exploratory Data Analysis in Python**", Green Tea Press, 2nd Edition, **2016**

Indicative Experiments

1. An agricultural data analyst is tasked with analyzing crop production data	2 hours
across various states in India to identify trends, distributions, and correlations. The dataset includes fields such as State, Crop, Area_Sown, Production, Yield and Year. The analyst is required to complete the following two tasks: 1. Identify and handle any missing or invalid values in the dataset and remove any duplicate records to ensure data integrity. 2. Calculate the mean, median, mode, standard deviation, and range for Area_Sown, Production, and Yield. Recommended to use any publicly available free datasets.	

2. A hospital data analyst is preparing patient admission records for performance monitoring. The dataset includes fields such as Admission_Type, Diagnosis_Code, Age, Gender, Total_Charges, and Length_of_Stay. Perform the following tasks: • Identify and remove duplicate admission records. • Handle missing or invalid values in Age and Total_Charges. • Group patients by Admission_Type and calculate the average length of stay and the mean total charges. • Reshape the dataset to determine which admission type has the highest cost-to-stay ratio. Recommended to use any publicly available free datasets.	2 hours
3. A bank is analyzing applicant data to predict loan approvals. The dataset includes fields such as Gender, Married, Applicant Income, and Loan_Status. Implement the following tasks: • Create age groups (e.g., 50) by binning the Age column. • Encode categorical variables like Gender and Married using dummy variables. • Demonstrate how dummy variables prevent numerical bias in algorithms such as Logistic Regression, and calculate the average loan approval rate for each age group. Recommended to use any publicly available free datasets.	2 hours
4. A data scientist is analyzing customer arrivals and purchase behavior in a retail store. To model various real-world scenarios, perform the following tasks for each distribution. • Generate random data samples. • Plot histograms and the corresponding probability density functions (PDFs) or probability mass functions (PMFs). • Compute the mean and variance of each distribution. Recommended to use any publicly available free datasets.	4 hours
5. A data analyst is working at a retail company wants to understand customer spending patterns using descriptive statistics. The dataset contains information on annual income, spending score, and age of customers.Implement variance, skewness, kurtosis, and percentile calculations for these variables to interpret customer behavior. 1. Compute the variance, skewness, and kurtosis for Age,	4 hours
Annual Income,and Spending Score. 2. Find the 25th, 50th, and 75th percentiles for each variable and Interpret the results what do these metrics indicate about customer distribution.Recommended to use any publicly available free datasets.	
6. A hotel manager wants to understand booking patterns and cancellations to improve operations. 1. Group data by hotel type (City/Resort)and calculate the average lead time and cancellation rate. 2. For each country,find the average number of adults and average total nights booked and identify the month with the highest number of bookings and cancellations. Recommended to use any publicly available free datasets.	4 hours

7. A marketing analyst wants to understand the relationships between customer characteristics and their purchasing behavior from a retail dataset. The dataset includes variables such as age, income, gender, number of purchases, and total spending. 1. Perform bivariate analysis to examine the relationship between age and total spending, to determine if older customers tend to spend more. 2. Analyze the association between gender and purchase frequency using appropriate bivariate techniques. Recommended to use any publicly available free datasets.	4 hours
8. A retail company wants to analyze its daily sales data over the last two years to understand sales trends, seasonal effects, and forecast future sales to optimize inventory management. Perform a time series analysis on the sales data. Recommended to use any publicly available free datasets.	4 hours
9. A regional wine retailer has compiled a dataset of various chemical properties for three classes of wines. The dataset includes features like alcohol content, malic acid, ash, and other measurements from lab analysis. Management wants to simplify this data and visualize the main differences between wine types for marketing and quality control purposes.Apply Principal Component Analysis to reduce the number of features while retaining most of the information. Recommended to use any publicly available free datasets.	2 hours
10. A data analytics team at Retail Mart Ltd. is tasked with exploring a large dataset of monthly sales transactions across its nationwide stores. Use Pandas Profiling, Sweetviz, and D-Tale to generate automated exploratory reports to identify data-type mismatches, outliers, and missing values. Employ advanced visualization techniques to gain deeper insight: Sankey diagram, Violin plot, Parallel Coordinates, Radial plots and Gauge charts.Recommended to use any publicly available free datasets.	2 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI408	Statistical Inference	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand the fundamental concepts of probability and statistical inference.
2. To apply hypothesis testing methods for analyzing population parameters and sample data.
3. To analyze relationships and dependencies in data using correlation and regression techniques.

Course Outcomes

1. Apply estimation techniques including point and interval estimation to real-world data.
2. Perform and interpret hypothesis testing for population parameters.
3. Analyze correlations, regressions and relationships in data.
4. Evaluate the reliability and validity of statistical inferences drawn from data.
5. Interpret statistical results to support evidence-based decision-making.

Module:1	**Fundamentals of Statistical Inference**	**9 hours**

Concepts and Definitions: Population, sample, parameter, and statistic. Estimators and Estimates: Properties of good estimatorsâ€”unbiasedness, consistency, invariance. Sufficiency: Factorization theorem, minimal sufficiency, and applications. Efficiency: Concepts of efficiency; Lehmann-ScheffÃ©'s theorem; Rao-Blackwell theorem. Bayesian Estimation: Prior, posterior distributions, and Bayesian principles. Error Types: Type I and II errors, power of a test, and minimizing errors.

Module:2	**Estimation Theory**	**9 hours**

Point Estimation Methods: Maximum Likelihood Estimation (MLE), method of moments, least squares, minimum chi-square. Large Sample Properties of MLE: Consistency and asymptotic normality (without proof). Comparative Analysis: Bias and efficiency of estimators. Confidence Intervals: For means,proportions, and variances. Duality: Link between hypothesis tests and confidence intervals.

Module:3	**Hypothesis Testing and Non parametric Methods**	**10 hours**

Neyman - Pearson Lemma and Most Powerful Tests, Uniformly Most Powerful Tests: Notion and properties, Likelihood Ratio Tests: Applications to standard distributions and composite hypotheses, Non-parametric Tests: Sign Test, Mann-Whitney, Wilcoxon signed-rank test. Bayesian Inference in Testing,Introduction to Multivariate Inference.

Module:4	**Regression and ANOVA**	**10 hours**

Simple and Multiple Linear Regression: Assumptions, coefficients, multicollinearity. Model Diagnostics: Residual analysis, leverage, influence measures (Cook's distance). Advanced Regression: Ridge regression, LASSO, Generalized Linear Models (GLMs). Analysis of Variance (ANOVA): One-way and two-way ANOVA - assumptions, interpretation, Post-hoc Tests: Tukey's HSD, Bonferroni correction, ScheffÃ©'s method. Mixed-Design ANOVA: Interactions in factorial designs.

Module:5	**Statistical Software Applications**	**5 hours**

Overview of Software: R, Python, SPSS; strengths and weaknesses in statistical analysis. Data Manipulation: Using dplyr (R) or pandas (Python) for data cleaning and transformation. Data Visualization: ggplot2 (R) or Matplotlib/Seaborn (Python); effective graphical representation of statistical data. Real-world Applications: Case studies and projects using real datasets for hands-on experience.

Module:6	**Contemporary Topics**	**2 hours**

	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining, "**Introduction to Linear Regression Analysis**", John Wiley and Sons Inc, 6th Edition, **2021**

Reference Books

1. Ronald Christensen, "**Analysis of Variance, Design, and Regression: Applied Statistical Methods**", CRC Press, 1st Edition, **1996**
2. Robert V. Hogg, Elliot A Tanis, Dale L. Zimmerman, "**Probability and Statistical Inference**", Pearson Education, 9th Edition, **2015**
3. Manoj Kumar Srivastava, Namita Srivastava, "**Statistical Inference: Testing of Hypotheses** ", PHI Learning Pvt Ltd, 1st Edition, **2009**
4. Ronald Christensen, " **Analysis of Variance, Design, and Regression: Linear Modeling for Unbalanced Data**", Chapman and Hall CRC Press, 2nd Edition, **2016**
5. Marc S. Paolella , "**Fundamental Statistical Inference: A Computational Approach**", Wiley, 1st Edition, **2018**

Indicative Experiments

1. Introduction to Statistical Softwares: Installation and setup (R/Python/SPSS)	2 hours
2. Basic Data Manipulation and Visualization Techniques	4 hours
3. Point Estimation and Construction of Confidence Intervals using Sample Data	4 hours
4. Conducting Various Hypothesis Tests (z-tests, t-tests, chi-square tests) using Datasets	4 hours
5. Applying Non-Parametric Tests to Real-World Data	2 hours
6. Fitting Simple and Multiple Regression Models	4 hours
7. Interpreting Regression Outputs	2 hours

8. Performing one-way and two-way ANOVA and interpreting results	4 hours
9. Students will select a dataset, perform a comprehensive analysis using statistical methods, and present their findings.	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI409	Predictive Analytics	3	1	0	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To introduce the concepts, models, and applications of predictive analytics for informed decision-making.
2. To prepare and transform data through exploratory data analysis and statistical techniques for effective model building.
3. To apply and evaluate predictive modelling approaches including regression, classification, ensemble methods, and time series forecasting.

Course Outcomes

1. Comprehend and apply the core principles of predictive analytics to derive actionable insights across diverse real-world domains.
2. Perform data preprocessing and exploratory data analysis to identify relationships, trends, and patterns in data.
3. Develop and evaluate predictive models using regression, classification and ensemble techniques.
4. Apply techniques for model tuning, handling class imbalance, uplift modeling and time series forecasting to build accurate and interpretable predictive models.
5. Implement predictive analytics applications in real-world domains such as healthcare, cyber security, marketing, and social media analytics.

Module:1	**Introduction**	**6 hours**

Predictive analytics in Action- Analytics landscape - Types of analytics - Analytical Techniques - Predictive modeling -Statistical modeling- Opportunities in Analytics -Tools and applications.

Module:2	**Data Preparation**	**9 hours**

Classification of data, Data preparation methods - Data transformations for individual and multiple predictors - Overfitting and Model Tuning - Descriptive Analytics(mean, median, mode) - Variance and distribution - Shape of distribution -Covariance and correlation - Variable reduction - Hypothesis testing -Chi-square - Fit statistics - Stochastic models.

Module:3	**Predictive Models for Regression and Classification**	**10 hours**

Measuring Performance in Regression Models- Linear Regression and its Cousins- Nonlinear Regression Models- Regression Trees and Rule-Based Models- Measuring Performance in Classification Models- Discriminant Analysis and Other Linear Classification Models- Nonlinear Classification Models- Classification Trees and Rule-Based Models.

Module:4	**Handling Class Imbalance and Time Series Analysis**	**9 hours**

The Effect of Class Imbalance - Model tuning - Alternate cutoffs - Adjusting prior probabilities - Unequal case weights - Sampling methods - Cost-sensitive training. Uplift Modeling - Measuring uplift- uplift modeling approach.Measuring predictor importance - Factors that can affect model performance-Methods for time series analyses - Prediction and forecasting - Autoregressive Model - ARMA, ARIMA, ARFIMA.		
Module:5	**Applications of Predictive Analytics**	**9 hours**
Healthcare Data Analytics - Electronic Health Records,Mining Sensor Data in Medical Informatics, Social Media Analysis for Detection and Tracking of Infectious Disease Outbreaks, Modelling Churn prediction system, Malware Data Analytics- Machine Learning based Intrusion Detection System.		
Module:6	**Contemporary Topics**	**2 hours**

Total Lecture Hours:	**45 hours**
Tutorial Hours:	**15 hours**

Text Book(s)

1. Max Kuhn, Kjell Johnson, "**Applied Predictive Modeling**", Springer, 1st Edition, **2018**
2. Richard V. McCarthy, Mary M. McCarthy, Wendy Ceccucci, Leila Halawi, "**Applying Predictive Analytics: Finding Value in Data**", Springer, 1st Edition, **2019**
3. Chandan K. Reddy, Charu C. Aggarwal , "**Healthcare Data Analytics**", CRC Press Taylor and Francis Group, 1st Edition, **2015**

Reference Books

1. Jeffrey Strickland, "**Predictive Analytics Using R**", Colorado Springs Lulu, 1st Edition, **2015**
2. Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, "**Mining of Massive Datasets**", Cambridge University Press, 3rd Edition, **2020**

Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test

Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI410	Healthcare Data Analytics	3	1	0	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand the fundamentals of healthcare data and the role of analytics in healthcare decision-making.
2. To explain the types, sources, and structures of healthcare data, including electronic health records and medical imaging data.
3. To apply data preprocessing, cleaning, and integration techniques specific to healthcare datasets.

Course Outcomes

1. Explain the fundamentals and importance of healthcare data analytics.
2. Analyze different types and sources of healthcare data, including electronic health records and medical imaging data.
3. Apply data preprocessing, cleaning, and integration techniques to healthcare datasets.
4. Develop and implement statistical and machine learning models to extract insights from healthcare data.
5. Interpret analytical results to support clinical, operational, and policy decisions.

Module:1	**Introduction to Healthcare Informatics and Data Standards**	**8 hours**

Basics of Healthcare Data & EHR Systems - Interoperability Standards: HL7, FHIR, ICD-10/11, SNOMED CT-Healthcare. Databases: Relational and NoSQL Systems. Case study: Building an interoperable patient registry.

Module:2	**Healthcare Data Engineering and Preprocessing**	**8 hours**

Healthcare data sources: EHR, claims, registries, IoT, imaging - Data cleaning: handling missing values, duplicates, and noise - ETL pipelines for healthcare data - Cloud-based storage and healthcare data lakes. Case study: Building a de-identified research database.

Module:3	**Machine Learning for Healthcare Analytics**	**9 hours**

Supervised learning and Unsupervised learning with real time applications. Clustering, Patient stratification - Predictive analytics: Readmissions, Mortality, Chronic disease outcomes - Evaluation metrics: AUROC,Precision-recall, Calibration. Case study: Predicting hospital readmissions and more.

Module:4	**Healthcare Applications and Visualization**	**9 hours**

Data visualization principles for healthcare - Dashboards with Power BI,Tableau, Python/R libraries - Healthcare quality measures: HVBP, HRR, HAC programs - IoT/wearable

applications in patient monitoring. Case study: Interactive dashboard for infection control monitoring and more.		
Module:5	**Trends in Healthcare Data Analytics**	**9 hours**
Federated and Privacy-Preserving Healthcare Analytics- Multimodal Health Data Fusion-Real-time and Streaming Health Analytics - Predictive & Prescriptive Analytics- Sustainability and Future Directions.		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**
	Tutorial Hours:	**15 hours**

Text Book(s)

1. Chandan K. Reddy, Charu C. Aggarwal, "**Healthcare Data Analytics** ", CRC Press A Chapman and Hall Book, 1st Edition, **2015**
2. Sergio Consoli, Diego Reforgiato Recupero, Milan PetkoviÄ‡, "**Data Science for Healthcare: Methodologies and Applications**", Springer Nature Switzerland AG, 1st Edition, **2019**
3. Eric Topol , "**Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again**", Basic Books, 1st Edition, **2019**

Reference Books

1. Robert E. Hoyt, Ann K. Yoshihashi, "**Health Informatics: Practical Guide for Healthcare and Information Technology Professionals**", Lulu dot com, 6th Edition, **2014**
2. Nilmini Wickramasinghe, "**Digital Health: A Primer**", Chapman and Hall CRC Press, 1st Edition, **2024**

Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI411	Financial Data Analytics	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand the fundamentals of financial data and the role of analytics in finance.
2. To analyze and articulate the different types, sources and structures of financial datasets.
3. To apply data preprocessing, cleaning and transformation techniques for financial data analysis

Course Outcomes

1. Analyze and categorize the types, sources and structures of financial datasets.
2. Apply data preprocessing, cleaning, and transformation techniques to prepare financial data for analysis.
3. Develop and implement statistical, machine learning and predictive models to extract insights from financial data.
4. Interpret analytical results to support investment, risk management and financial decision-making
5. Evaluate the accuracy, reliability and effectiveness of financial analytics models and techniques

Module:1	**Finance Analytics and Linear Time Series Analysis**	**8 hours**

Analytics in Finance - Data-Driven Finance - Use of R and Python in Finance - Data in Finance: Obtaining Fundamental Data- Market Data. Foundation Analytics in Finance - Data-Driven Finance - Use of R and Python in Finance - Data in Finance: Obtaining Fundamental Data- Market Data. Foundation of Time Series: Stationarity, Correlation and Auto Correlation Function of Time Series: Stationarity, Correlation and Auto Correlation Function

Module:2	**Linear models for financial time series**	**9 hours**

Linear Time Series - Simple Autoregressive Models -Properties of AR Models - Simple Moving Average Models - Properties of MA Models - Simple ARMA Models - Properties of MA Models - General ARMA Models, Unit Root Non Stationarity

Module:3	**Seasonal and Long memory models**	**8 hours**

Exponential Smoothing - Seasonal Models - Seasonal Differencing - Multiplicative Seasonal Models - Regression Models with Time Series Errors -Consistent Covariance Matrix Estimation - Long Memory Models

Module:4	**Conditional Heteroscedastic and Continuous Time Models**	**9 hours**

Characteristics of Volatility - Structure of a Model - Model Building: Testing for ARCH Effect - ARCH Model- GARCH Model - Integrated GARCH Model - GARCH-M Model - Exponential GARCH Model - Threshold GARCH Model - Stochastic Volatility Model - Alternative Approaches**.** Continuous-Time Stochastic Processes - ITO'S Lemma - Distributions of Stock

Prices and Log Returns- Black-Scholes Pricing Formulas-Stochastic Integral- Jump Fusion Models		
Module:5	**Applications in Finance**	**9 hours**
Predicting Stock Returns: Accounting Ratios - Predicting Stock Returns: Prices - Past Stock Returns Data: Random Walk Model-Use of RNN and LSTM-Use of CNN-ARIMA Model-Neural Network Model. Sourcing Text Data- Text Preprocessing- Case Study: Summarizing a Document		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Pitabas Mohanty , "**Financial Analytics**", Wiley Publisher, 1st Edition, **2023**
2. Ruey S. Tsay, "**An Introduction to Analysis of Financial Data with R**", Wiley, 1st Edition, **2013**

Reference Books

1. Sam Chen, Ka Chun Cheung, Phillip Yam, "**Financial Data Analytics with Machine Learning, Optimization and Statistics**", Wiley, 1st Edition, **2023**
2. Ruey S. Tsay, "**Analysis of Financial Time Series**", Wiley, 3rd Edition, **2010**
3. William G. Foote, "**Financial Engineering Analytics: A Practice Manual Using R**", Wiley, 1st Edition, **2018**

Indicative Experiments

1. Using the Daily Returns dataset, perform the given statistical analysis of the daily returns to understand its volatility and risk characteristics. Import the financial dataset into the chosen data analysis environment (such as Python or Excel) for further processing and analysis. Compute the following descriptive statistics for the Daily_Return (%) column: • Standard Deviation • Skewness • Excess Kurtosis • Minimum • Maximum Based on the computed statistics, determine whether the stock demonstrates normal, high-risk (volatile), or skewed return behavior. Recommended to use any publicly available free datasets.	4 hours
2. Perform exploratory data analysis on financial stock price datasets to compute summary statistics, study trends, assess volatility, detect anomalies, and visualize data using charts such as line plots, histograms, and correlation matrices. Recommended to use any publicly available free datasets.	2 hours
3. Build and evaluate time series models (ARMA and ARIMA) on historical stock price data to predict future financial outcomes such as stock prices and	4 hours

returns. Analyze model performance, visualize predicted versus actual values, and interpret results to understand market trends and volatility. Historical Data of NIFTY-50 index from NSE India: Recommended to use any publicly available free datasets	
4. Consider the financial data containing monthly mortgage rates, Build a time series model for the conditional heteroscedasticity in the data. Apply ARCH and GARCH models to capture time-varying volatility. Perform model diagnostics to check the adequacy of the fitted models. Financial data set containing monthly mortgage rates: Recommendedto use any publicly available free datasets.	4 hours
5. Build a predictive model for detecting fraudulent transactions in a Synthetic financial dataset using simple linear regression and multilinear regression model. Synthetic financial dataset: Recommended to use any publicly available free datasets	2 hours
6. Use financial loan data set and build a logisticregression model that predicts whether a borrower will default on a loan basedon various financial indicators and demographic information. Financial LoanData Set: Recommendedto use any publicly available free datasets.	2 hours
7. Using the S&P 500 stock dataset, which provides historical prices of all companies listed from 2018 to 2023, create an optimal investment portfolio. Apply Modern Portfolio Theory (MPT) and financial data analysis techniques to maximize returns while minimizing risk. S&P 500 stock dataset: Recommended to use any publicly available free datasets.	4 hours
8. Leverage historical daily OHLC and volume data from the Yahoo Finance Dataset (2018-2023) to forecast asset prices and compute expected returns using neural network techniques. Yahoo Finance Dataset (2018-2023): Recommended to use any publicly available free datasets.	2 hours
9. Classify financial news articles from the Sentiment Analysis for Financial News dataset into positive, negative, or neutral categories, and study the correlation between news sentiment and stock price changes. Financial News dataset: Recommended to use any publicly available free datasets.	4 hours
10. Apply Natural Language Processing (NLP) techniques toanalyze text data from emails, chat logs, and transaction records to identifysuspicious activities or potential regulatory violations. The Enron EmailDataset. Recommendedto use any publicly available free datasets	2 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI412	Advanced Data Visualization Techniques	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand the principles and importance of advanced data visualization in data-driven decision-making.
2. To apply interactive and dynamic visualization techniques to explore and analyze data.
3. To design dashboards and visual interfaces to communicate insights effectively.

Course Outcomes

1. Identify the different data types, visualization types to bring out the insight.
2. Identify and create various visualizations for scalar, vector, hierarchical, network and screening data.
3. Apply exploratory data analysis and visualization techniques on different datasets using appropriate tools and techniques.
4. Apply and create visualization for diverse types of data.
5. Implement advanced visualization principles for creating an interactive dash board to support decision making and monitoring live data.

Module:1	**Introduction to Data Visualization**	**6 hours**

Overview of Data Visualization - Value of Visualization - Data Abstraction - Task Abstraction - Analysis: Four Levels for Validation -. Visualization Techniques -Scalar and point techniques - Colour maps - Contouring - Height Plots - Vector visualization techniques - Vector properties - Vector Glyphs - Vector Color Coding

Module:2	**Visual Encoding**	**7 hours**

Data Models - Conceptual Models - Dataset types - Attribute types - Visual Encoding - Visual Variables- Networks and Trees -Tables - Map Color - Manipulate View

Module:3	**Data modeling and Loading Techniques**	**10 hours**

Overview of QlikView data modeling - Associative data model - Facts, dimensions, and star schema design - Managing keys and associations - Synthetic and composite keys - Joins and concatenations - Data mapping using ApplyMap. Data extraction and transformation - QVD and QVX file usage - ETL scripting - Incremental and optimized data loading - Performance tuning for visualization efficiency

Module:4	**Diverse Types of Visual Analysis**	**10 hours**

Introduction to GIS - Spatial Analysis - Feature Representation - GIS Data Management - Statistical maps - Chloropleth map, Dot map, Symbol map, cartogram map- Types - Time-Series data visualization - Text data visualization - Matrix visualization techniques - Multivariate data visualization - Narrative visualization

Module:5	**Advanced Visualization**	**10 hours**

Introduction to Data Streaming - processing and presenting of streaming data - streaming visualization techniques - streaming analysis. Fundamentals of interactive dashboard creation, Dashboard creation using visualization tools for the use cases: Finance- marketing- insurance- healthcare - Challenges, Limitations and Advantages		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. GarcÃa, Miguel Ã ngel, "**QlikView: Advanced Data Visualization: Discover deeper insights with Qlikview by building your own rich analytical applications from scratch**", Packt, 1st Edition, **2018**
2. Alexandru C. Telea, "**Data Visualization: Principles and Practice**", CRC Press, 1st Edition, **2014**
3. Mathew Ward, Georges Grinstein and Daniel Keim, "**Interactive Data Visualization: Foundations, Techniques and Applications**", A K Peters , 1st Edition, **2015**

Reference Books

1. Brunsdon, C., and L. Comber, L, "**An Introduction to R for Spatial Analysis and Mapping**", SAGE Publications Ltd, 2nd Edition, **2018**
2. Juuso Koponen, Jonatan Hilden, "**Data Visualization Handbook**", Aalto yliopisto, 1st Edition, **2018**
3. Anthony Aragues , "**Visualizing Streaming Data: Interactive Analysis Beyond Static Limits**", OReilly, 1st Edition, **2018**
4. Ryan Sleeper, "**Practical Tableau: 100 Tips, Tutorials, and Strategies from a Tableau Zen Master**", OReilly, 1st Edition, **2018**

Indicative Experiments

1. Analyze if a marketing strategy is efficient using different plots.	2 hours
2. An e-commerce company wants to understand the distribution of customer ratings for a newly launched product to improve quality and customer satisfaction. Create distribution visualizations that show how ratings are spread across customers and identify common trends or issues.	4 hours
3. A company manages a large office building and wants to monitor room temperatures across different floors in real time to ensure comfort and energy efficiency. Create scalar data visualizations that show temperature values and detect anomalies in specific rooms or areas.	2 hours
4. **Hierarchical Clustering** A retail company wants to understand patterns among its customers to improve marketing strategies. They've collected data on customer behavior, including: Annual spending, Purchase frequency, Product categories preferred, Loyalty score. Group similar customers together based on their purchasing behavior and visualize the same.	4 hours
5. A social media analytics company wants to analyze how information spreads across a social network like Twitter (X) or Instagram. They collect data on users, their followers, and how they interact (likes, comments, shares, mentions). Use network visualization to visualize the interaction between	2 hours

users.	
6. A city's transportation department wants to analyze and monitor traffic patterns to reduce congestion and improve road safety. Create a spatial data visualization to help identify the congestion zones.	2 hours
7. A city's energy department wants to analyze electricity usage trends across different districts to manage power distribution efficiently and reduce wastage. Create a time series visualization that shows how electricity consumption changes over time (hourly, daily, monthly, and seasonally.	4 hours
8. A news analytics company wants to understand public sentiment about a newly introduced government policy by analyzing thousands of tweets and Facebook comments. Create text data visualizations that reveal key themes, opinions, and emotional tones expressed by the public.	2 hours
9. A hospital wants to analyze multiple health indicators simultaneously to identify patterns that could help in predicting diseases such as diabetes or heart problems. Create multivariate data visualizations that show how these factors interact and which combinations correlate with certain health risks.	2 hours
10. Assume a dataset containing the details of professors such as their name, designation, years of experience, years since phd, salary and so on. Visualize the professors according to their designation, salary and years of experience.	2 hours
11. A logistics company manages hundreds of delivery trucks operating across different cities. They need to monitor their fleet in real time to ensure timely deliveries, driver safety, and efficient route management. Create a real-time (streaming) data visualization dashboard that displays the live status and performance of all vehicles.	2 hours
12. A university's academic department wants to monitor student performance across different programs and semesters to improve teaching strategies and identify struggling students early. Design an interactive dashboard that gives administrators, faculty, and advisors a clear overview of academic performance and trends.	2 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI413	Advanced Data Compression Techniques	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To understand the basic idea of Advanced data compression techniques.
2. To understand the various compression technique used for text, image and Video.
3. To obtain insight on various compression algorithms for current real-world problems.

Course Outcomes

1. Understand the basic mathematical concepts and information theory used for data compression.
2. Design and analyze the different approaches for lossless data compression.
3. Comprehend the various approaches for lossy data compression.
4. Analyze the various image and video compression techniques for multimedia application.
5. Design the methods and tools for wavelet-based compression.

Module:1	**Fundamentals of Compression and Information Theory**	**10 hours**

Introduction to Compression techniques - Mathematical preliminaries for Lossless compression - Entropy - Information Value - Concepts of information theory - Models and Coding - Physical Models - Probability models - Markov models - Applications of compression

Module:2	**Lossless Coding Techniques**	**8 hours**

Shannon-Fano Algorithm - Huffman Coding Algorithm - Canonical Huffman codes - Adaptive Huffman Coding - Golomb codes - Rice codes - Tunstall codes- Application of Huffman coding for text compression

Module:3	**Dictionary methods and Standards**	**7 hours**

Dictionary Methods: LZ77, LZ78, LZW, LZSS, LZMA and LZ4 Algorithms - Lossless Compression standards zip, Unix compress, GIF, JBIG, PPM

Module:4	**Lossy Coding Quantization and Image Compression**	**10 hours**

Basics of lossy coding and mathematical concepts - Distortion criteria - Scalar quantization - Uniform quantizer - Adaptive quantization - Advantages of vector quantization over scalar quantization - LBG algorithm - Image Compression: Discrete Cosine Transform - JPEG2000 - JPEG-LS

Module:5	**Audio and Video Compression**	**8 hours**

Audio Compression: Run-Length Encoding - Voc File Compression - Linear Predictive Coding - The Foundation: Psychoacoustics. Video Compression: Motion Compensation - Temporal and Spatial Prediction - MPEG-2 - H.264		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**
Text Book(s)		
1. Khalid Sayood, "**Introduction to Data Compression**", Morgan Kaufmann, 5th Edition, **2017**		
Reference Books		
1. Colton McAnlis, Aleks Haecky, "**Understanding Compression: Data Compression for Modern Developers**", OReilly, 1st Edition, **2016** 2. Feng Wu, "**Advances in Visual Data Compression and Communication Meeting the Requirements of New Applications**", Auerbach, 1st Edition, **2014**		
Indicative Experiments		
1. Implement Shannon-Fano coding for given data set, encode and decode it and calculate the compression ratio. Demonstrates statistical coding principles in data compression		2 hours
2. Implement a program that takes a data stream as input and performs one-pass lossless compression by dynamically building and updating the Huffman tree as each symbol is processed. • Encode the input data stream using Adaptive Huffman coding. • Decode the compressed output to reconstruct the original data. • Verify that the decoded data matches the original input. Recommended to use any publicly available free datasets		4 hours
3. Implement the Lempel-Ziv (LZ78) algorithm for adaptive variable-length encoding to compress and decompress the given dataset and evaluate its compression efficiency. Recommended to use any publicly available free datasets		2 hours
4. Implement the Rice Coding algorithm to compress and decompress the given OEIS dataset using adaptive variable-length encoding. The program should accept the dataset as input and perform encoding for different values of the parameter kkk (where M=2kM = 2^kM=2k). For each dataset, generate the encoded bitstream, decode it back to the original sequence and verify the correctness of the decoding process. Finally, compute the original size, compressed size, and compression ratio for each case, and analyze how the choice of kkk affects the compression efficiency. Recommended to use any publicly available free datasets		4 hours
5. Design a program that takes a Medical Dataset as input and applies LZ78 and LZW compression.		4 hours

• Encode the input using each algorithm, building the dictionary dynamically. • Decode the compressed output to reconstruct the original message. • Verify that decoding matches the original input. • Compute and compare original size, compressed size, and compression ratio for each algorithm. • Observe and analyze how the dictionary grows during encoding and how repetitive patterns in the data affect compression efficiency. Recommended to use any publicly available free datasets.	
6. Implement a uniform quantizes for an analog signal. Compute the quantized signal, quantization error and MSE. Plot the original and quantized signals along with the error, and analyze how increasing the number of quantization bits affects signal quality. Recommended to use any publicly available free datasets	2 hours
7. Apply the LBG algorithm to a dataset of vectors to generate a codebook of the desired size. In each iteration, assign every input vector to its nearest code vector, compute the quantization error (distortion), and update the code vectors. Repeat the process until convergence, i.e., when the change in distortion between successive iterations is below a specified threshold. Recommended to use any publicly available free datasets.	2 hours
8. Apply the Discrete Cosine Transform (DCT) to a given datasetâ€"such as a signal, image, or multidimensional dataâ€"to analyse its frequency components, observe energy compaction, and evaluate reconstruction quality. • Compute the DCT of the input data to obtain frequency-domain coefficients. • Optionally, truncate or threshold selected coefficients to simulate compression. • Perform the Inverse DCT (IDCT) to reconstruct the original data. • Compare the original and reconstructed data using metrics like Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR). Recommended to use any publicly available free datasets.	4 hours
9. Compress and decompress a VOC audio file to analyze compression efficiency and its effect on audio quality. Recommended to use any publicly available free datasets	3 hours
10. Analyse and Compress an audio signal using Linear Predictive Coding (LPC) and evaluate reconstruction quality. Recommended to use any publicly available free datasets.	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Course Based Design Project, Course Project	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IAMDI414	Computer Vision	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To identify major image analysis approach in computer vision
2. To provide thorough understanding of digital image processing for computer vision applications
3. To demonstrate and design a typical computer vision for real time problems

Course Outcomes

1. Comprehend image formation principles and camera systems
2. Evaluate and apply feature extraction techniques
3. Implement 3D vision techniques for real-world problems
4. Analyze and develop methods of understanding the significance of classifiers in vision-based applications
5. Design and innovate real-world computer vision applications

Module:1	Image Formation and Camera Geometry	8 hours

Introduction to Computer Vision - Image Formation - Radiometry- Sources, Shadows and Shading Color - Representing Colors - Surface Colour from Image Colour - Geometric Image Features - Elements of Analytical Euclidean Geometry - Geometric Camera Parameters - Calibration Methods - Analytical Image Features

Module:2	Feature Extraction	9 hours

Linear Filters and Convolution - Shift Invariant Linear Systems - Spatial Frequency and Fourier Transforms - Sampling and Aliasing - Edge Detection - Filters and Features - Texture - The Geometry of Multiple Views

Module:3	3D Vision and Motion Analysis	9 hours

3D Vision Techniques - Stereopsis - Affine Structure from Motion - Projective Structure from Motion - Segmentation Using Clustering Methods - Fitting - Segmentation and Fitting Using Probabilistic Methods - Tracking

Module:4	Object Recognition and Classification	8 hours

Correspondence and Pose Consistency - Finding Templates Using Classifiers - Recognition by Relations between Templates - Aspect Graphs - Range Data - Finding in Digital Libraries

Module:5	Computer Vision Applications	9 hours

Image-Based Rendering - Biometric Systems - Face Recognition - Vision-based Precision Agriculture - Industrial Applications - Medical Imaging Applications - Autonomous Vehicle Vision - Current Research Trends - Vision Ethics

Module:6	Contemporary Topics	2 hours
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. D. A Forsyth, "**Computer Vision a Modern Approach**", Pearson Education, 2nd Edition, **2015**

Reference Books

1. R.Szeliski, "**Computer Vision: Algorithms and Applications**", Springer, 2nd Edition, **2022**
2. Katsushi Ikeuchi, "**Computer Vision: A Reference Guide**", Springer, 2nd Edition, **2021**
3. Richard Hartley and Andrew Zisserman, "**Multiple View Geometry in Computer Vision**", Cambridge University Press, 2nd Edition, **2004**
4. S.Khan, H.Rahmani, S.Shah, M.Bennamoun, "**A Guide to Convolutional Neural Networks for Computer Vision**", Morgan and Claypool Publishers, 1st Edition, **2018**

Indicative Experiments

1. **Basic Image Processing Operations** Enhancing satellite imagery for crop health monitoring in precision agriculture. • Applying histogram equalization and contrast adjustment to multi spectral farm images • **Dataset:** Agriculture-Vision Dataset. Recommended to use any publicly available free datasets. Preprocessing underwater images for marine biology research. • Color correction and dehazing for coral reef monitoring • **Dataset:** SUIM Underwater Image Dataset. Recommended to use any publicly available free datasets.	3 hours
2. **Image Enhancement and Alignment Operations** Detecting potholes and road damage in smart city infrastructure. • Edge enhancement and morphological operations for road surface analysis • **Dataset:** RDD2020 Road Damage Dataset. Recommended to use any publicly available free datasets. Image alignment for panoramic stitching in real estate virtual tours. • Feature-based image registration and blending • **Dataset:** [Custom real estate images or Flickr panorama dataset]. Recommended to use any publicly available free datasets.	3 hours
3. **Image Segmentation and Feature Extraction** Identifying defective products on manufacturing assembly lines.	3 hours

• Segmenting product components and extracting texture features for quality control • **Dataset:** MVTec AD Anomaly Detection Dataset. Recommended to use any publicly available free datasets. Segmenting fire and smoke regions in forest surveillance systems. • Color-based segmentation and region growing for wildfire detection • **Dataset:** FLAME Fire Dataset. Recommended to use any publicly available free datasets.	
4. **Object Detection using YOLO, RCNN** Detecting and counting vehicles in traffic management systems. • Real-time vehicle detection for traffic density analysis and parking lot monitoring • **Dataset:** KITTI Vision Benchmark or UA-DETRAC. Recommended to use any publicly available free datasets. Detecting protective equipment (helmets, vests) in construction site safety monitoring. • Object detection for worker safety compliance verification • **Dataset:** Hard Hat Detection Dataset. Recommended to use any publicly available free datasets.	3 hours
5. **Object Tracking using Kalman Filter** Tracking ball trajectory in sports analytics (soccer/basketball). • Predicting ball position and velocity for game strategy analysis • **Dataset:** Sports-1M Dataset or SoccerNet. Recommended to use any publicly available free datasets. Tracking wildlife animals in conservation monitoring systems. • Multi-object tracking for animal behavior studies in natural habitats • **Dataset:** Wildlife Detection Dataset or ENA24 Dataset	3 hours
6. **3D Reconstruction** Creating 3D models of archaeological artifacts for digital preservation. • Structure from Motion (SfM) for heritage documentation • **Dataset:** [Custom archaeological images or Sketchfab cultural heritage models] Depth estimation for indoor navigation systems for visually impaired individuals. • Monocular depth estimation for obstacle detection and path planning • **Dataset:** NYU Depth V2 Dataset or ScanNet	3 hours

7. **Facial Analysis and Recognition** Implementing emotion detection for customer sentiment analysis in retail. • Facial landmark detection and emotion classification • **Dataset:** FER2013 Dataset or AffectNet	3 hours
8. **Gesture Recognition** Hand gesture recognition for touchless human-computer interaction. • Real-time hand tracking and gesture classification for smart home control • **Dataset:** EgoGesture Dataset or 20BN-Jester Dataset. Recommended to use any publicly available free datasets.	3 hours
9. **Motion Analysis and Activity Recognition** Detecting falls and abnormal activities in elderly care monitoring. • Temporal motion pattern analysis for healthcare applications • **Dataset:** UR Fall Detection Dataset or NTU RGB+D Dataset	3 hours
10. **Image Classification using CNNs** Classifying different types of recyclable waste for smart waste management. • Deep learning-based classification for automated sorting systems • **Dataset:** TrashNet Dataset or Waste Classification Dataset	3 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI404	Deep Learning	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To introduce major deep neural network frameworks and challenges in basic neural networks
2. To discuss design and development of an artificial neural network for classification
3. To give an insight on real world applications using Deep Learning

Course Outcomes

1. Understand the core mechanics of deep neural networks, model evaluation deployment and scalability
2. Implement regularization, optimization algorithm and model enhancement approaches for robust model development
3. Illustrate the concepts of convolutional neural networks and pre-trained models for problem solving
4. Develop real-world applications using Recurrent Neural Networks
5. Apply advanced architectures such as Generative Adversarial Networks and attention based learning for real world applications

Module:1	**Introduction To Deep Learning**	**6 hours**

Introduction to Deep Learning- Biological Neural Networks- Artificial Neural Networks: Forward Propagation - Gradient Descent- Backpropagation- Activation Functions -BERT and GPT Architecture- Confusion Matrix - ROC-Model Deployment and Scalability

Module:2	**Optimization of Deep Learning Models**	**7 hours**

Setup and Initialization Issues - Hyperparameters Tuning- Feature Engineering - Generalization issues - underfitting vs overfitting - The Vanishing and Exploding Gradient Problems - Gradient Descent Intuitions - Different Types of Gradient Descent - Exponential Weighted Averages - Gradient Descent with Momentum - RMSProp, AdaGrad and Adam Optimization- dropout - Batch Normalization - Softmax Regression

Module:3	**Convolutional Networks**	**10 hours**

Basic Structure of convolution network - The Convolution Operation -Motivation -Pooling - Convolution and Pooling as an Infinitely Strong Prior -Variants of the Basic Convolution Function - Structured Outputs -Data Types -Efficient Convolution Algorithms - Random or Unsupervised Features - CNNs Architectures- ResNet, AlexNet, InceptionNet and MobileNets - CNN Application - Content-Based Image Retrieval - Object Localization

Module:4	**Sequence Modeling Recurrent And Recursive Nets**	**10 hours**

Recurrent Neural Networks -Architecture - Bidirectional RNNs - Encoder-Decoder Sequence-to-Sequence Architectures - Deep Recurrent Networks -Recursive Neural Network - The Challenge of Long-Term Dependencies - Echo State Networks - Long Short-Term Memory and Other Gated RNNs		
Module:5	**Generative Adversarial Networks**	**10 hours**
Generative Modelling - Need and Definition - Autoencoders - Variational autoencoders; Generative Adversarial Networks (GANs) - Generator and discriminator roles - Training dynamics and adversarial loss - Types: vanilla GAN, Conditional GAN (CGAN), Deep Convolutional GAN (DCGAN), Wasserstein GAN (WGAN); Attention Mechanism: Attention Cues, Attention Pooling, Scoring Functions, Self-Attention and Positional Encoding- GAN Application - automatic image captioning and Image generation		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Ian Goodfellow Yoshua Bengio Aaron Courville, "**Deep Learning**", MIT Press, 1st Edition, **2017**
2. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J, "**Dive into Deep Learning**", Cambridge University Press, 2nd Edition, **2023**
3. Charu C. Aggarwal, "**Neural Networks and Deep Learning: A Textbook**", Springer, 2nd Edition, **2023**

Reference Books

1. Nikhil Buduma and Nicholas Locasio, "**Fundamentals of Deep Learning**", O Reilly, 2nd Edition, **2022**
2. Josh Patterson, Adam Gibson, "**Deep Learning: A Practitioner's Approach**", O Reilly Media, 1st Edition, **2017**
3. Bishop, C. M., & Bishop, H, "**Deep learning: Foundations and Concepts**", Springer Nature, 1st Edition, **2023**

Indicative Experiments

1. Assume you're tasked with designing a neural network to simulate an XOR gate that accepts four binary inputs. Using TensorFlow, create an Artificial Neural Network (ANN)model to learn the XOR function. Train the model for 10 epochs and report relevant metrics such as loss, accuracy, precision, and recall. After training, use your model to predict the output for the input combination [1, 1, 1, 0] and explain how the network arrived at this decision.	2 hours
2. A retail company, ShopSmart Pvt. Ltd., maintains historical data of its monthly product sales over the past 5 years. The management wants to forecast the next 12 months of sales to make better inventory and marketing decisions. You are assigned to develop a time series forecasting model using LSTM, as traditional regression models failed to capture sequential dependencies. Reference Dataset: Retail Sales Dataset on Kaggle Repository.	2 hours

3. A company named VisionX Robotics Pvt. Ltd. develops automated warehouse robots that rely on camera-based object detection to identify items (like boxes, bottles, and tools) on conveyor belts. However, the dataset of item images is limited. The company wants to Detect and classify objects in existing images. Generate new synthetic images of similar objects using GANs, to improve model training and performance. You are tasked with demonstrating a lab experiment that combines object detection and GAN-based image generation. Reference Dataset: Inventory Warehouse Object Detection Dataset on Kaggle Repository.	4 hours
4. A company named Secure Vision Technologies is developing a smart attendance system for corporate offices and universities. The system should automatically recognize employees or students using a webcam or stored face images. You are assigned to design a CNN-based face recognition model that can classify faces of known individuals accurately. Reference Dataset: LFW Dataset on Kaggle Repository.	4 hours
5. You are working as a data scientist in a publishing company that wants to automate content creation for blogs, product descriptions, and social media posts. The goal is to generate high-quality, contextually relevant text using a generative AI model while ensuring diversity and coherence. Reference Dataset: AI vs Human Text Dataset on Kaggle Repository.	4 hours
6. You are a bioinformatics researcher analyzing DNA sequences to identify potential gene patterns associated with hereditary diseases. The dataset contains sequences of nucleotides (A, T, C, G), and your task is to predict certain motifs or anomalies that may indicate genetic disorders using Recurrent Neural Networks(RNN).	4 hours
7. You are a machine learning engineer tasked with designing a Deep Neural Network to solve the classic XOR problem. The XOR function cannot be solved using a simple linear model, so your goal is to implement a DNN that can accurately learn the non-linear relationship between inputs and outputs. Reference Dataset: Exclusive XOR Dataset on Kaggle Repository	2 hours
8. Your task is to train three different models to classify the images: an Artificial Neural Network(ANN), a Convolutional Neural Network (CNN), and a pre-trained model (such as MobileNet or VGG16). Reference: Tensorflow datasets	4 hours
9. You are tasked with developing a classification model to assess the quality of African plums based on their visual appearance. Reference to African Plums Quality dataset that contains images that fall into several quality categories but suffers from a class imbalance issue, with certain categories significantly underrepresented. To tackle this problem, you will fine-tune the last 10 layers of a pre-trainedInceptionV3 model. Additionally, apply appropriate strategies to address class imbalance, such as data augmentation techniques,	4 hours
oversampling, or undersampling. After fine-tuning the model, evaluate its performance using key classification metrics, such as accuracy, precision, recall, F1 score, and AUC (Area Under the Curve).	
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	

Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI405	MLOps	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To Learn MLOps fundamentals and different tools for implementation
2. To enable building production-ready, scalable, and responsible ML systems by mastering MLOps principles like lifecycle management, versioning, deployment, and monitoring
3. To develop the skills to precisely explore and implement advanced applications and generative models with a focused consideration of ethical and governance aspects

Course Outcomes

1. Understand MLOps process and tools
2. Develop and automate ML pipelines for testing, developing, and deploying models
3. Leverage tools for largescale model management
4. Design responsible AI systems with reproducibility, reliability, and ethical deployment
5. Evaluate bias, ensure fairness, and govern AI ethically for responsible deployment

Module:1	**Fundamentals of Machine Learning and MLOps**	**8 hours**

Introduction to ML - Supervised vs. Unsupervised Learning, Bias-variance tradeoff, Model Evaluation Metrics. Introduction to MLOps - Definition, importance, lifecycle stages, and key challenges. Overview of tools and cloud platforms (AWS, Google Cloud AI, IBM Watson)

Module:2	**Versioning Pipelines and Preprocessing**	**10 hours**

Version control using Git and DVC for datasets. Model versioning using MLflow. Pipeline orchestration with Kubeflow. Data preprocessing and validation for ML and NLP tasks - Tokenization, TF-IDF, Embeddings, Byte Pair Encoding

Module:3	**Continuous Integration Deployment Strategies and Transformers**	**10 hours**

CI/CD fundamentals for ML systems. Deployment strategies: batch, online, and streaming. Serving models with FastAPI/Flask, Docker, Kubernetes, and serverless platforms (AWS Lambda, Google Cloud Functions). Introduction to Transformers - BERT, GPT models, and generative techniques like GANs and Diffusion Models

Module:4	**LLMOPs and Advanced Applications**	**9 hours**

Introduction to LLMOPs and tools like LangChain and OpenAI API. Prompt engineering, Retrieval-Augmented Generation (RAG), and Low-Rank Adaptation (LoRA). Real-world applications in finance, healthcare, retail, and smart cities

Module:5	**Monitoring Governance and Ethical AI**	**6 hours**

Model degradation and drift detection techniques. Feedback loops and retraining pipelines. Governance, compliance, and responsible MLOps practices		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. Mark Treveil and Alok Shukla, "**Introducing MLOps: How to Scale Machine Learning Projects with Continuous Delivery, Monitoring, and Automation**", O Reilly, 1st Edition, **2020**

Reference Books

1. Yaron Haviv and Noah Gift, "**Implementing MLOps in the Enterprise**", O Reilly, 1st Edition, **2023**
2. Noah Gift and Alfredo Deza, "**Practical MLOps: Operationalizing Machine Learning odels**", O Reilly, 1st Edition, **2021**
3. Hannes Hapke and Catherine Nelson, "**Building Machine Learning Pipelines**", O Reilly, 1st Edition, **2020**

Indicative Experiments

1. Design a small ML project fully reproducible using Git, DVC, and Kubeflow pipelines	2 hours
2. Track experiments with MLflow by logging params/metrics/artifacts and register the best model	2 hours
3. Build a verifiable ML data pipeline in Kubeflow with automated validation in CI/CD	2 hours
4. Publish features to a feature store using Feast on Kubernetes and demonstrate train vs. online parity	2 hours
5. Package a trained MLflow model into a Docker container and expose REST API with FastAPI	2 hours
6. Deploy the containerized model on Kubernetes and perform a canary update.	4 hours
7. Instrument Kubernetes inference service with Prometheus/Grafana and set dashboards and alert rule	4 hours

8. A Customer Conversational Intelligence Platform is a system that employs advanced technologies, including machine learning and natural language processing, to analyze and make sense of customer interactions across various communication channels such as chatbots, call centers, emails, and social media that modern businesses accumulate. The experiment will develop a platform that harnesses the power of machine learning to analyze vast amounts of customer interaction data. The aim is to derive actionable insights from these interactions, optimize customer service processes, and enhance overall customer experience. The dataset such as 3K Conversations Dataset for ChatBot, Customer Support on Twitter Dataset can be collected from Kaggle Repository for developing the Customer Conversational Intelligence Platform.	4 hours
9. This experiment aims to develop a robust solution for detecting financial fraud using MLOps methodologies. It involves analyzing historical transaction data, customer behavior, and relevant factors to identify and prevent fraudulent activities in the financial domain. The financial sector faces a growing threat from financial fraud, driven by increasingly sophisticated tactics used by fraudsters. This experiment addresses the need for advanced fraud detection solutions capable of adapting to emerging threats in real-time using the Paysim synthetic dataset collected from Kaggle Repository	4 hours
10. The field of finance can be complex and overwhelming for individuals seeking personalized financial advice. In order to make informed decisions regarding investments, retirement planning, budgeting, and financial products, individuals often require guidance from financial experts. The aim of this experiment is to develop an Intelligent Financial Advisor powered by a Large Language Model (LLM) to provide personalized financial advice and guidance to individuals. By leveraging NLP and machine learning techniques, the Intelligent Financial Advisor will assist users in making informed financial decisions and achieving their financial goals	4 hours
Total Laboratory Hours:	**30 hours**
Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI406	Gen AI and Large Language Models	3	0	2	4
Pre-requisite	NIL	**Syllabus Version**			
		1			

Course Objectives

1. To provide a foundational understanding of Generative AI and Large Language Models
2. To equip students with knowledge of core mechanisms in LLMs
3. To develop proficiency to design, train and leveraging LLMs

Course Outcomes

1. Understand the language representation techniques to build and evaluate generative and representation models
2. Apply transformer architecture and diffusion models for natural language and vision-based generative tasks
3. Design and implement prompt engineering strategies to improve generative model outputs
4. Analyze the functioning of multimodal large language models and evaluate using standard performance metrics
5. Develop a new application by applying fine-tuning techniques on embedding-based language models

Module:1	**Introduction to GenAI**	**7 hours**

Representation Learning, Unsupervised Deep Learning, Autoencoders Latent Space Representation, Variational Autoencoders, Generative Adversarial Networks: Architecture, Training Process, Challenges, CycleGAN

Module:2	**Transformers**	**9 hours**

Word Embeddings, Attention mechanism, Transformers, LlaMA, BERT, Text-to-Text Transfer Transformer, GPT, Vision Transformers, Diffusion models- Forward Diffusion Process, Reverse Denoising Process, Stable diffusion

Module:3	**Prompt Engineering**	**9 hours**

Prompting Techniques -Zero-shot, One-shot, Few-shot Prompting, Chain prompting, Chain-of-Thought, Self-consistency, Tree - of- Thought, Role Prompts and Task Templates, Advanced text generation techniques: Model I/O, Chains, Memory, Agents, Semantic search and RAG

Module:4	**Multimodal Large Language Models**	**9 hours**

Transformer for Vision, Multimodal embedding models-CLIP, Making text generation models multimodal-BLIP, Evaluation metrics - Inception Score, Frechet Inception Distance, CLIP, Perplexity, BLEU, METEOR

Module:5	**Fine Tuning Language Models**	**9 hours**

Embedding models, SBERT, Train model, loss functions, Fine-tuning Representation Models for Classification, Fine-tuning Generation Models, instruction tuning with QLoRA, Evaluating generative models, RLHF		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

1. David Foster, "**Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play**", O Reilly Media, 2nd Edition, **2023**
2. Jay Alammar, Maarten Grootendorst, "**Hands-On Large Language Models**", O Reilly Media, 1st Edition, **2024**

Reference Books

1. Danis Rothman, "**Transformers for Natural Language Processing and Computer Vision**", Packt Publishing, 3rd Edition, **2024**
2. SavaÅŸ YÄ±ldÄ±rÄ±m, Meysam Asgari-Chenaghlu, "**Mastering Transformers**", Packt Publishing, 1st Edition, **2021**

Indicative Experiments

1. Train a classifier on the MNIST digits dataset using a discriminative and generative model and compare their performance. Reference to MNIST Dataset from Public Repositories	2 hours
2. Implement unpaired image-to-image translation on Horse2Zebra dataset using CycleGAN. Reference toHorse2zebra Dataset from Public Repositories	2 hours
3. Generate image using stable diffusion and evaluate the model. Reference to Wikiart dataset on KaggleData Repository	2 hours
4. Train Word2Vec on a small dataset and visualize word analogies using Text8 Corpus. Reference to Text8 - Word Embedding available on Kaggle repository	2 hours
5. Classify IMDb movie reviews as positive/negative using T5. Reference to IMDB Dataset of 50K Movie Reviews available onPublic Data Repositories	2 hours
6. Compare LLM performance for sentiment classification under different prompting styles. Reference to IMDB Dataset of 50K Movie Reviews available on Public Data Repositories	4 hours
7. Build a retrieval-augmented generation system (use FAISS and LangChain) to answer domain-specific queries. Refer to MedQuAD available on Kaggle data repository	2 hours
8. Classify CIFAR-10 dataset with ViT and compare with CNN baseline. The CIFAR-10 dataset is available on multiple platforms with public access	2 hours

9. Generate image captions using BLIP/Gemini and Compute BLEU, METEOR for machine translation, Inception Score, FID for generated images, CLIP score for text-image alignment. Reference to Flickr Image captioning dataset available on Kaggle	4 hours
10. Develop a plagiarism detection tool to check semantic similarity between student essays and online sources. a. Fine-tune SBERT on a Semantic Textual Similarity (STS) dataset b. Compare cosine similarity between sentence pairs before and after fine-tuning c. Visualize sentence clusters with t-SNE Refer to Semantic Textual Similarity (STS) dataset available on GitHub repository with public access	8 hours
Total Laboratory Hours:	**30 hours**

Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar

Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025

Course Code	Course Title	L	T	P	C
IACSI407	Agentic AI	3	0	2	4

Pre-requisite	NIL	**Syllabus Version**
		1

Course Objectives

To define intelligent agents, multi-agent architecture and Agentic-AI framework
To provide knowledge on creating an LLM based reasoning agent for Agentic- Agent Systems design
To analyze the challenges and failure modes of agents

Course Outcomes

Understand the fundamentals of intelligent agents and its types
Design an LLM based reasoning agent for an Agentic-AI application Development
Build and evaluate agentic systems
Explore the working of multi-agents in Agentic-AI framework
Analyze and assess the ethical and responsive aspects in Agentic AI framework

Module:1	**Introduction to Agentic AI**	**7 hours**

Intelligent Agents and their characteristics - history and evolution of agentic AI - Understanding self-governance, agency, and autonomy - architecture of agentic systems - Deliberative - Reactive - Hybrid - Reasoning in Intelligent agents - Decision making and planning in agentic systems

Module:2	**Prompt Engineering**	**9 hours**

Basics of Transformer architecture, tokenization, context windows, hallucinations, and bias - Prompting - Zero-Shot, Few-Shot Prompting - Chain-of-Thought (CoT) and Tree-of-Thought (ToT) - Short-Term Memory - Long-Term Memory - Retrieval- Augmented Generation (RAG) - Fine-tuning and PEFT (LoRA, adapters)		
Module:3	**Multimodal Applications and Tool Integration**	**9 hours**
Working with Chat models - Reasoning Models - Controlling model behaviour - Running models locally - Multimodal AI applications - Text to image - DALL-E - Using stable diffusion - Image Understanding - Concept of Tool Use - ReACT- Built-in langchain tools - Custom tools - Advanced tool calling capabilities - Incorporating tools into workflows		
Module:4	**Multi Agent Systems and Collaboration**	**9 hours**
Introduction to Multi-Agent Systems - Coordinator-Worker-Delegator (CWD) - CWD model for the intelligent travel agent - Designing agents with role assignments - Roles and responsibilities of each agent - Communication and collaboration between agents - Implementing the CWD approach in generative AI systems - CrewAI, AutoGen, LanGraph - Model Context Protocols (MCP)		
Module:5	**Building and Evaluating Agentic Systems**	**9 hours**
Security: Preventing prompt injection, ensuring safe tool execution - Operational metrics for LLM applications - Ethical and Responsible Design of Agentic AI - Societal Implications - Economic distribution and equity considerations - Indian AI Policy Frameworks		
Module:6	**Contemporary Topics**	**2 hours**
	Total Lecture Hours:	**45 hours**

Text Book(s)

Anjanava Biswas and Wrick Talukdar, "**Building Agentic AI Systems: Create intelligent, autonomous AI agents that can reason, plan, and adapt**", Packt Publishing, 1st Edition, **2025**
Louis-FranÃ§ois Bouchard and Louie Peters, "**Building LLMs for Production: Enhancing LLM Abilities and Reliability with Prompting, Fine-Tuning, and RAG**", O Reilly Towards AI Series, 2nd Edition, **2024**

Reference Books

Ben Auffarth, Leonid Kuligin, "**Build production-ready LLM applications and advanced agents using Python, LangChain, and LangGraph**", Packt Publishers, 2nd Edition, **2025**
Steve Wilson, "**The Developer's Playbook for Large Language Model Security**", O Reilly Media Inc, 1st Edition, **2024**
Lewis Tunstall, Leandro von Werra, and Thomas Wolf, "**Natural Language Processing with Transformers: Building Language Applications with Hugging Face**", O Reilly Media Inc, 1st Edition, **2022**
Sinan Ozdemir, "**Quick Start Guide to Large Language Models: Strategies and Best Practices for ChatGPT, Embeddings, Fine-Tuning, and Multimodal AI**", O Reilly Media Inc, 2nd Edition, **2024**

Indicative Experiments

1. Note:Laboratory exercises shall be implemented using a combination of contemporary agentic AI frameworks and workflow orchestration tools, including LangChain, LangGraph, CrewAI, and AutoGen, along with n8n for workflow automation and external tool integration. Hello World of Agentic AI using Langchain - Use ReAct paradigm that can solve math word problems by deciding when to use a calculator tool.	2 hours
2. Build a Research Assistant with Web Search - Create an agent that answers open-ended questions by formulating search queries.	2 hours
3. Retrieval-Augmented Generation (RAG) Agent - Load a custom document (e.g., a PDF of a research paper or a company handbook), create a vector store, and build an agent that answers questions based only on the provided document	2 hours
4. Code Generation and Execution Agent - Build an agent that takes a natural language request (e.g., "create a list of the first 10 prime numbers"), writes Python code, and executes it in a safe, sandboxed environment.	2 hours
5. Build a Stateful Agent with LangGraph - Create an agent with a built-in feedback loop that can "reflect" on its answer and self- correct if the initial output is poor.	4 hours
6. Automated Task Breakdown with Hierarchical Planning - Implement an	2 hours
AutoGPT-style agent that takes a high-level goal and autonomously breaks it down into sub-tasks executing them sequentially.	
7. Supervisor-Worker Model with CrewAI - Create a CrewAI crew with a "Research Manager" agent that delegates tasks to a "Researcher" agent and a "Writer" agent to produce a short report on a given topic	2 hours
8. Collaborative Debate with AutoGen - Set up two AutoGen agents with different personas (e.g., an "Optimist" and a "Pessimist") and have them debate the pros and cons of a topic. Use a third "Moderator" agent to facilitate and summarize the discussion	2 hours
9. Evaluating Agent Performance on a Benchmark - Run a simple agent from previous experiments on a set of 20 predefined questions, a mini-benchmark. Develop metrics for success rate, number of steps taken, and cost (number of tokens used) to quantitatively evaluate its performance.	2 hours
10. Evaluating Agent Performance on a Benchmark - Run a simple agent from previous experiments on a set of 20 predefined questions, a mini-benchmark. Develop metrics for success rate, number of steps taken, and cost (number of tokens used) to quantitatively evaluate its performance.	2 hours
11. Build a Multi-Modal Agent - analyze images and answer natural language questions about their content by combining visual perception models and LLM-based reasoning	4 hours
12. Dynamic Multi-Agent Workflows with LangGraph - Model a complex approval workflow (e.g., a travel request) using LangGraph	4 hours
Total Laboratory Hours:	**30 hours**

Mode of Evaluation: Continuous Assessment Test, Digital Assignment, Quiz, Final Assessment Test, Lab Continuous Assessment, Lab Final Assessment, Seminar	
Recommended by Board of Studies :	16-10-2025
Approved by Academic Council : No. 80	20-11-2025