SNDT Women's University, Mumbai

Undergraduate Degree / UG Programme (Syllabus as Per NEP) - Faculty of Science & Technology

Bachelor of Science (Artificial Intelligence - AEDP)

B.Sc. In - AI AEDP

As Per NEP – 2020

Semester – I to VI

Syllabus (W.E.F. Academic Year 2025-26)

Terminologies

Vertical	Full-form/Definition	Remarks	Related to Major and Minor Courses
Major (Core)	Subject comprising Mandatory and Elective Courses, Major Specific IKS, Vocational Skill Courses, Internship/ Apprenticeship, Field Projects, Research Projects connected to Major	Minimum 50% of total credits corresponding to Three/Four - year UG Degree- Mandatory Courses	Related to the Major
Minor Course	Course from same or different Faculty	Minimum 18-20 Credits to be completed in the first three years of UG Programme	Related to the Minor
OEC	Open Elective Courses/ Generic courses	10-12 credits to be offered in I and/or II year. Faculty-wise baskets of OEC to be prepared	OEC is to be chosen compulsorily from faculty other than that of the Major
VSC	Vocational Skill Courses, including Hands on Training corresponding to the Major and/or Minor Subject	8-10 credits, to be offered in first three years, wherever applicable vocational courses will include skills based on advanced laboratory practical's of Major	Related to the Major or Minor
SEC	Skill Enhancement Courses	06 credits, to be offered in I and II year, to be selected from the basket of Skill Courses approved by university	Related to the Major or Minor any relevant Skill
AEC	Ability Enhancement Courses	08 credits, to be offered in I and II year, English: 04 Credits to be earned in Sem - I, Modern Indian Language of 04 credits to be offered in II year	NA

VEC	Value Education Courses	Understanding India, Environmental science/education, Digital and technological solutions, Health & Wellness, Yoga education, sports, and fitness	NA
IKS	Indian Knowledge System	Generic IKS Course: basic knowledge of the IKS to be offered at First Year level	Major-Specific IKS Courses: advanced information about the major, part of the major credit to be offered at second- or third-year level
OJT	On-Job Training (Internship/Apprenticeship)	Corresponding to the Major Subject	Related to the Major
FP	Field projects	Corresponding to the Major Subject	Related to the Major
CC	Co-curricular Courses	Health and Wellness, Yoga education sports, and fitness, Cultural Activities, NSS/NCC and Fine/ Applied/Visual/ Performing Arts	NA
CE	Community Engagement and service		Related to Major
RP	Research Project	corresponding to the Major Subject	Related to Major

Program Template

Programme		B.Sc.
Specialization		Artificial Intelligence
Programme Outcomes (POs) **Action Verbs demonstrating**		After completing this programme, Learner will
	1.	Demonstrate a strong foundation in Artificial Intelligence concepts, including knowledge of programming languages, data structures, mathematics for AI, machine learning fundamentals, and intelligent systems.
(Major) discipline-related knowledge acquisition, mastery over cognitive and professional, vocational skills are to be used e.g. demonstrate sound understanding of, analyses, compare, create, design, etc...(minimum5)	2.	Analyze ethical, social, and professional responsibilities in the field of Artificial Intelligence by evaluating the impact of AI systems on individuals, organizations, and society, with reference to professional standards and best practices.
	3.	Apply AI and programming knowledge to design and develop intelligent applications that solve real-world problems using techniques such as machine learning, deep learning, and data analytics.
	4.	Evaluate AI models, algorithms, and system architectures with respect to performance, scalability, security, fairness, and user experience.
	5.	Design AI-based software solutions that meet industrial standards, ensuring reliability, interpretability, and ethical compliance in intelligent systems.
	6.	To understand Data Analytics.
	7.	To Apply real world Problems.
Eligibility Criteria for Programme		1. B. Sc.IT – I Year: A women candidate must have passed the higher secondary school certificate (std. XII) examination conducted by the different divisional boards of the Maharashtra state board of secondary and higher secondary education in any stream with 45% marks (40% for candidates belonging to reserved category). Or must have passed an examination of any other recognized board or body recognized as equivalent. Students who have not done mathematics at 12th std. Are needed to take a bridge course in mathematics and pass in university conducted test before semester 1 examination. Or must have passed any three-year government recognized diploma programme in second class.

		2. B.Sc. (IT) - II Year: For a student from our university - should have cleared or has ATKT as admissible in first year in the same subject. For a student from another university- studied at least two courses of four credits each in the I year in the same subject. Passed I year without ATKT. Passed three-year full-time Diploma in Engineering with an aggregate of not less than 60% (Open Category) and 55% (Reserved Category) OR must have passed any three-year Diploma in Engineering Course. 3. B.Sc. IT - III Year: For a student from our University- should have cleared second year in the same subject or has passed with admissible ATKT. For a student from another university– should have completed at least six courses of 4 credits each or 24 credits in the I and II year in the same subject. Passed in I and II year without ATKT.
Intake (For SNDTWU Departments and Conducted Colleges)		As Per University Affiliation Letter

Structure with Course Titles

AEDP - B. Sc. In Artificial Intelligence

Semester - I

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester I					
10135511	Introduction to Logic & Reasoning (Th)	Major (Core)	4	100	50	50
10135502	Applied Linear Algebra (Th)	Major (Core)	2	50	50	0
10435511	Computer Vision- Concepts and Applications	OEC **(Any One)**	4	100	50	50
10435512	Understanding Artificial Intelligence					
10435513	Artificial Intelligence: Basics and Applications					
10635501	Computational Thinking and Problem Solving	VSC S1 **(Related to Major)**	2	50	50	0
	Swayam / Chetana / MOOC	SEC **(Any One)**	2	50	0	50
10735511	Introduction To operating System					
10735512	GitHub & Version Control					
10810111	English For Academic Writing - Paper I (For Students of English Medium)	AEC (English) **(Any One)**	2	50	0	50
10810112	English Language and Literature - I (For Students of Non-English medium)					

	Ability Enhancement Course (AEC) Link: https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/aec-syllabus/ug-degree/ability-enhancement-course.pdf (Available on Website)					
11051111	Inception of India Knowledge System **IKS Link:** https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/iks-syllabus/ug-degree/inception-of-indian-knowledge-system.pdf (Available on Website)	IKS (Generic)	2	50	0	50
10952111	Introduction to Indian Constitution **Link:** https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/vec-syllabus/ug-degree/introduction-to-indian-constitution.pdf (Available on Website)	VEC	2	50	0	50
11450121	Basics of National Service Scheme	CC **(Any One)**	2	50	50	0
11450221	National Cadets Corps. (NCC) Studies – I					
11450322	Health and Wellness					
11450421	Performing Arts Exploration					

	Co-Curricular Course (CC) **Link:** https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/cc-syllabus/ug-degree/co-curricular-course-as-per-nep-2020-semester-i-syllabus.pdf (Available on Website)					
			22	**550**	**250**	**300**

Semester – II

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester II					
20135511	Python Programming	Major (Core)	4	100	50	50
20135512	Discrete Mathematics for AI	Major (Core)	2	50	0	50
20635501	Introduction to AI	VSC S2 **(Related to Major)**	2	50	50	0
20635502	Computer Organization and Architecture	VSC S3 **(Related to Minor)**	2	50	50	0
20435511	AI in Everyday Life	OEC **(Any One)**	4	100	50	50
20435512	Digital Intelligence and Society					
20435513	Spreadsheets and Power BI					
20735501	Linux operating system	SEC	2	50	50	0
20810111	English For Academic Writing - Paper II (For Students of English Medium)	AEC (English) **(Any One)**	2	50	0	50
20810112	English Language and Literature - II (For Students of Non-English medium)					
	Ability Enhancement Course (AEC) Link: https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/aec-syllabus/ug-degree/ability-					

	enhancement-course.pdf (Available on Website)					
20952111	Environment Awareness **Link:** https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/vec-syllabus/ug-degree/environment-awareness.pdf (Available on Website)	VEC	2	50	0	50
21450121	Volunteerism and National Service Scheme	CC **(Any One)**	2	50	50	0
21450221	National Cadets Corps. (NCC) Studies - II					
21450323	Yoga Education					
21450421	Fine Art					
	Co-Curricular Course (CC) Link: https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/cc-syllabus/ug-degree/co-curricular-course-as-per-nep-2020-semester-ii-syllabus.pdf (Available on Website)					
			22	**550**	**300**	**250**

Semester – III

SN	Courses	Type of Course	Credits	Marks	Int	Ext
30135511	Numerical Methods	Major (Core)	4	100	50	50
30135512	Data Structures and Algorithms	Major (Core)	4	100	50	50
30135513	Introduction to Data Science	Major (Core)	2	50	0	50
30335511	Data Analytics	Minor Stream	4	100	50	50
30435511	AI and the Future of the work	OEC **(Any One)**	2	50	0	50
30435512	Presentation Tools and AI					
30435513	Social and Ethical Issues in Artificial Intelligence					
	Modern Indian Language **Ability Enhancement Course (AEC) Link:** https://sndt.ac.in/pdf/academics/syllabus-as-per-nep/aec-syllabus/ug-degree/aec-semester-iii.pdf (Available on Website)	AEC **(Any One)**	2	50	50	0
30810301	रचनात्मक लेखन (Hindi)					
30810401	मराठी भाषेचा परिचय - भाग १ (Marathi)					
30810501	Contemporary Sanskrit Nyaya (Sanskrit)					
30810201	શીખો ગુજરાતી – પ્રાથમિક ભાગ ૧: લિપિ પરિચય, શ્રવણ અને વાચન					

	કૌશલ્ય (Gujarati)					
30135504	Cognitive Science for AI	Major (Core)	2	50	50	0
	Co-Curricular Course (CC) Link: https://sndt.ac.in/pdf/academics/syllabus-as-per-nep/cc-syllabus/ug-degree/co-curricular-course-as-per-nep-2020-semester-iii-syllabus.pdf (Available on Website)	CC **(Any One)**	2	50	50	0
31450121	Social issues Advocacy and Action					
31450221	National Cadets Corps. (NCC) Studies – III					
31450321	Traditional Sports and Fitness					
31450421	Unfolding The Beauty of Indian Music					
			22	**550**	**300**	**250**

Semester – IV

SN	Courses	Type of Course	Credits	Marks	Int	Ext
40135511	Database Management System	Major (Core)	4	100	50	50
40135512	Web Technology	Major (Core)	4	100	50	50
40335511	Machine learning	Minor Stream	4	100	50	50
40435511	AI in Healthcare	OEC **(Any One)**	2	50	0	50
40435512	Future Trends in Artificial Intelligence					
40435513	Data Literacy for Artificial Intelligence					
40735511	AI Safety, Privacy & Security	SEC	2	50	0	50
	Modern Indian Language **Ability Enhancement Course (AEC) Link:** https://sndt.ac.in/pdf/academics/syllabus-as-per-nep/aec-syllabus/ug-degree/aec-semester-iv.pdf (Available on Website)	AEC **(Any One)**	2	50	0	50
40810411	मराठी भाषेचा परिचय - भाग २ (Marathi)					
40810411	सूचना प्रौद्योगिकी और हिंदी भाषा (Hindi)					
40810511	वाल्मिकीकिरामयणे अयोध्याकाण्ड: (Sanskrit)					

40810211	શીખો ગુજરાતી - ભાધ્મમિક (Gujarati)					
40335502	Data Visualization and Data Analytics	Minor Stream	2	50	50	0
	Co-Curricular Course (CC) Link: https://sndt.ac.in/pdf/academics/syllabus-as-per-nep/cc-syllabus/ug-degree/co-curricular-course-as-per-nep-2020-semester-iv-syllabus.pdf (Available on Website)	CC **(Any One)**	2	50	50	0
41450122	Personality and Leadership Development through National Service Scheme					
41450121	NSS Volunteers under National service scheme special camp					
41450221	National Cadets Corps. (NCC) Studies – IV					
41450421	Theatre & Dance					
41450321	Basics of Self Defence and Fitness Training					
			22	**550**	**250**	**300**

Semester – V

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Explainable AI (XAI) and Model Interpretability	Major (Core)	4	100	50	50
	Generative AI and Applications	Major (Core)	4	100	50	50
	Indian Knowledge Systems and Ethical Practices in IT	IKS (Major Specific)	2	50	0	50
	Advanced Analytics with AI (Predictive & Prescriptive Analytics)	Minor Stream	4	100	50	50
	Deep learning	Minor Stream	4	100	50	50
	Big data Analytics	VSC-4	2	50	50	0
	Human-Centered AI	Major (Core)	2	50	50	0
			22	**550**	**300**	**250**

Semester – VI (AEDP)

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Internship (AEDP-Apprenticeship Embedded Degree Programmes)	Major (Core)	20	500	250	250
	Digital Literacy and Community Outreach	CE	2	50	0	50
			22	**550**	**250**	**300**

The minor Degree program introduced from the academic year 2025-26 are:
1. Data Analysis in AI

Details about the subject for titles for these one Minor degree courses are as given below.

Year and Term	Academic Semester	Data Analysis in AI	Credits
First year s first term	First	Applied Linear Algebra	2
First year second term	Second	Computer Organisation and Architecture	2
Second Year First Term	Third	Data Analytics	4
Second Year First Term	Fourth	Machine learning	4
Third Year First Term	Fourth	Data Visualization and Data Analytics	2
Fourth Year Second Term	Fifth	Advanced Analytics with AI (Predictive & Prescriptive Analytics)	4
Fourth Year First Term	Fifth	Deep learning	4
		Total	22

Course Syllabus

Semester - I

.1.1 Major (Core)

Course Title	**Introduction to Logic & Reasoning (Th)**
Course Credits	**4 Credits**
Course Outcomes	1. Understand principles of logical reasoning and argument structure.
	2. Apply propositional and predicate logic for AI reasoning.
	3. Analyze arguments for validity, soundness, and fallacies.
	4. Solve reasoning-based problems using symbolic methods.
	5. Use logic for decision-making in computational problems.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Define logic, propositions, and statements.
	2. Construct truth tables for logical expressions.
	3. Apply logical connectives in reasoning.
	4. Identify tautologies and contradictions.
Content Outline	• **Concepts**: Introduction to logic, statements, propositions, truth values. • **Logical Connectives**: AND, OR, NOT, implication, equivalence. • **Laws of Logic**: De Morgan's laws, tautologies, contradictions.

Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Differentiate between propositional and predicate logic.
	2. Use quantifiers in logical statements.
	3. Translate English sentences to logic expressions.
	4. Apply logic for decision-making problems.
Content Outline	• **Propositional Logic**: Syntax, semantics, applications. • **Predicate Logic**: Quantifiers (universal, existential), predicates. • **Translation**: Converting natural language to logical form.
Module 3 (Credit 1)	**After learning the module, learners will be able to**
Learning Outcomes	1. Analyze arguments for validity and soundness.
	2. Identify logical fallacies.
	3. Construct proofs using inference rules.
	4. Apply reasoning to computational contexts.
Content Outline	• **Reasoning**: Deductive, inductive reasoning. • **Fallacies**: Common fallacies in reasoning. • **Inference**: Rules of inference, validity testing.
Module 4 (Credit 1)	**After learning the module, learners will be able to**
Learning Outcomes	1. Apply logical reasoning to problem-solving.
	2. Use logic in decision-making models.
	3. Solve logic puzzles systematically.
	4. Relate reasoning to AI applications.

Content Outline	• **Applications**: Logic puzzles, reasoning games. • **AI Applications**: Role of logic in AI inference engines. • **Decision Models**: Rule-based decision-making.
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: • Create truth tables for given logical expressions. • Simplify logical expressions using De Morgan's laws. • Identify tautologies and contradictions from examples.	
Module 2: • Translate English sentences into propositional logic. • Apply quantifiers to represent conditions. • Group activity: Identify predicates for a real-world domain.	
Module 3 • Validate arguments using truth tables. • Identify fallacies from given examples. • Create proofs using rules of inference.	
Module 4 • Solve reasoning-based puzzles in groups. • Apply logic to solve decision problems. • Case study: AI system using logic-based decision-making.	

Reference Books

1. Copi, I. M., Cohen, C., & McMahon, K. (2018). *Introduction to Logic* (14th ed.). Pearson.
2. Suppes, P. (2019). *Introduction to Logic*. Dover.
3. Smith, P. (2020). *An Introduction to Formal Logic*. Cambridge University Press.
4. Lemmon, E. J. (2018). *Beginning Logic*. Chapman & Hall.
5. Barwise, J., & Etchemendy, J. (2011). *Language, Proof, and Logic*. CSLI.
6. Hurley, P. (2016). *A Concise Introduction to Logic*. Cengage.

Assessment:

Internal Assessment – 50 Marks

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Problem Understanding	● Identifies logical structure accurately	10
Application of Logic	● Uses correct logical methods	15
Solution Accuracy	● Correct conclusions derived	10
Presentation & Clarity	● Clear explanation of solutions	5
Class Activities	● Active participation in logic exercises	10

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: Logic Puzzle Solver – Create a reasoning-based solver in Python that validates truth values.

Project 2: Mini Expert System – Develop a simple rule-based system that applies logical inference for recommendations.

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted by the college

Semester - I

.1.2 Major (Core)

Course Title	**Applied Linear Algebra (Th)**
Course Credits	**2 Credits**
Course Outcomes	1. Understand the role of linear algebra in AI and ML.
	2. Perform operations on matrices and vectors.
	3. Solve systems of linear equations for AI applications.
	4. Apply determinants and inverse matrices in computations.
	5. Understand eigenvalues and eigenvectors in AI algorithms (e.g., PCA).
	6. Use linear algebra concepts for dimensionality reduction and optimization.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Define vectors and matrices.
	2. Perform basic vector and matrix operations.
	3. Apply operations to AI-related problems.
Content Outline	• **Vectors**: Scalars, vectors, vector addition, scalar multiplication. • **Matrices**: Types (square, diagonal, identity), matrix addition, subtraction, multiplication. • **Matrix Transpose**: Symmetric and skew-symmetric matrices. • **AI Application**: Data representation as matrices in ML datasets.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Calculate determinants for 2×2, 3×3 matrices.

	2. Find inverse of a matrix.
	3. Apply determinant properties in AI computations.
Content Outline	• **Determinants**: Expansion by minors, properties. • **Inverse Matrices**: Adjoint method, existence of inverse. • **Applications**: Solving linear systems, transformation matrices in computer vision.
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: • Represent image pixel data as a matrix. • Perform addition and multiplication on sample matrices. • Transpose a dataset matrix and explain changes.	
Module 2: • Compute determinant of a 3×3 matrix. • Find inverse of a given matrix using adjoint method. • Apply inverse matrix to solve transformation problem.	

Reference books:

1. Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley.
2. Lay, D. C. (2015). *Linear Algebra and Its Applications*. Pearson.
3. Kolman, B., & Hill, D. R. (2013). *Elementary Linear Algebra*. Pearson.
4. Anton, H., & Rorres, C. (2014). *Elementary Linear Algebra with Applications*. Wiley.
5. Hefferon, J. (2020). *Linear Algebra*. Open Textbook.
6. Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra*. SIAM.

Internal Assessment – 50 Marks

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Matrix Operations	Correct application of matrix arithmetic	10
Determinants & Inverse	Correct computation and application	10
Solving Linear Equations	Accuracy in solving and interpreting results	10
Eigenvalues & PCA	Correct calculation and application to datasets	15
Presentation & Documentation	Clear explanation of solutions	5

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: PCA on Dataset – Reduce dimensions of an AI dataset using eigen decomposition in Python.
Project 2: Image Compression – Apply singular value decomposition (SVD) to compress an image dataset.

References book:

1. Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist*. O'Reilly.
2. Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly.
3. Sweigart, A. (2019). *Automate the Boring Stuff with Python* (2nd ed.). No Starch Press.
4. Zelle, J. (2017). *Python Programming: An Introduction to Computer Science*. Franklin.
5. Beazley, D. (2009). *Python Essential Reference*. Addison-Wesley.
6. Ramalho, L. (2015). *Fluent Python*. O'Reilly.

Assessment:

Internal Assessment: (50 marks) Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Code Implementation	Correct syntax, structure, and execution	15
Problem Solving Approach	Logical problem-solving methods applied	10
Output Correctness	Program produces expected output	10
Code Quality	Indentation, variable names, comments	5
Debugging Skills	Ability to fix syntax/logical errors	5
Lab Record/Assignments	Completeness and correctness	5

Given below are two sample projects but it is expected to work on similar sort of projects.

Project 1: Student Grade Calculator – Python program to calculate grades based on marks, with input validation.

Project 2: Basic Data Analysis Script – Program to load CSV file and compute mean, median, and max marks using Python.

Semester - I

.1.3 A. Open Elective Courses/ Generic (OEC)

Course Title	**Computer Vision – Concepts and Applications**
Course Credits	**4 Credits**
Course Outcomes	After going through the course, learners will be able to
	1. Describe the entrepreneurship process.
	2. Explain the types and role of entrepreneurs
	3. Discuss the various Government Schemes for entrepreneur development schemes
	4. Case studies of successful entrepreneurs
Module 1(Credit 1)	**Fundamentals of Computer Vision and Image Processing**
Learning Outcomes	After learning the module, learners will be able to
	1. Explain the scope and applications of computer vision.
	2. Understand image representation and color models.
	3. Perform basic image processing operations.
Content Outline	➢ **Introduction to Computer Vision** • Human vision vs computer vision • Applications: healthcare, surveillance, autonomous vehicles, AR/VR ➢ **Image Formation and Representation** • Digital images, pixels, resolution • Sampling and quantization ➢ **Color Spaces and Models** • RGB, Grayscale, HSV, CMY
Module 2 (Credit 1)	**Feature Detection, Extraction, and Image Analysis**
Learning Outcomes	After learning the module, learners will be able to
	1. Detect important features from images.

	2. Analyze edges, corners, and regions.
	3. Apply feature extraction techniques for object understanding.
Content Outline	➢ **Corner and Interest Point Detection** • Harris corner detector • Shi–Tomasi detector ➢ **Feature Descriptors** • SIFT, SURF, ORB (conceptual overview) ➢ **Image Segmentation** • Thresholding techniques • Region-based segmentation
Module 3 (Credit 1)	**Object Detection, Recognition, and Motion Analysis**
Learning Outcomes	After learning the module, learners will be able to
	1. Understand object detection and recognition techniques.
	2. Apply template matching and classification methods.
	3. Analyze motion and track objects in videos.
	4. Implement basic vision-based detection systems.
Content Outline	➢ **Object Recognition Techniques** • Feature-based recognition • Classical classifiers overview ➢ **Introduction to Face Detection** • Haar cascades • Applications and limitations ➢ **Motion Analysis** • Frame differencing • Optical flow (basic concept) ➢ **Object Tracking** • Tracking-by-detection approach

Module 4 (Credit 1) Deep Learning for Computer Vision and Applications	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand the role of deep learning in computer vision.
	2. Explain convolutional neural networks (CNNs).
	3. Apply pretrained models for vision tasks.
	4. Analyze real-world computer vision applications.
Content Outline	➢ **Introduction to Deep Learning for Vision** • Limitations of traditional vision methods ➢ **Convolutional Neural Networks (CNNs)** • Convolution, pooling, and activation functions • CNN architecture overview ➢ **Popular CNN Architectures (Conceptual)** • LeNet, AlexNet, VGG, ResNet ➢ **Transfer Learning** • Use of pretrained models ➢ **Vision Applications** • Image classification • Face recognition • Medical image analysis • Autonomous driving

Semester - I

.1.3 B. Open Elective Courses/ Generic (OEC)

Course Title	**Understanding Artificial Intelligence**
Course Credits	**4 Credits**
Course Outcomes	After going through the course, learners will be able to
	1. Explain the basic concepts, history, and importance of Artificial Intelligence.
	2. Identify different types of AI and their working principles.
	3. Understand how AI is used in daily life and various industries.
	4. Analyze ethical, social, and future impacts of Artificial Intelligence.
Module 1(Credit 1): Introduction to Artificial Intelligence	
Learning Outcomes	After learning the module, learners will be able to
	1. Define Artificial Intelligence in simple terms.
	2. Differentiate between human intelligence and machine intelligence.
	3. Identify the evolution and milestones of AI.
Content Outline	➢ **What is Artificial Intelligence?** ➢ **Intelligence: Human vs Machine** ➢ **History and Evolution of AI** 1. Early AI ideas 2. Growth of AI in modern times ➢ **Examples of AI in Daily Life** • Google search, voice assistants

Module 2(Credit 1): How Artificial Intelligence Works (Basic Concepts)	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand how machines learn from data.
	2. Identify basic AI techniques without technical complexity.
Content Outline	➢ **Data and Information** • Importance of data in AI • Types of data (text, image, audio, video) ➢ **Introduction to Machine Learning** • Learning from examples • Supervised and unsupervised learning (basic idea)
Module 3(Credit 1): Applications of Artificial Intelligence	
Learning Outcomes	After learning the module, learners will be able to
	1. Identify AI applications in various fields.
	2. Understand how AI improves efficiency and decision-making.
	3. Analyze real-life AI use cases.
Content Outline	➢ **AI in Daily Life** • Smart phones and smart devices • Chatbots and virtual assistants ➢ **AI in Education** • Personalized learning • Online assessment systems ➢ **AI in Healthcare** • Disease prediction • Medical imaging
Module 4(Credit 1): Ethics, Challenges, and Future of Artificial Intelligence	
Learning Outcomes	After learning the module, learners will be able to

	1. Understand ethical issues related to AI.
	2. Identify challenges and limitations of AI systems.
Content Outline	➢ **Ethical Issues in AI** • Bias and fairness • Privacy and data security ➢ **Social Impact of AI** • Job automation and employment • Human–AI collaboration

Semester - I

.1.3 C. Open Elective Courses/ Generic (OEC)

Course Title	**Artificial Intelligence: Basics and Applications**
Course Credits	**4 Credits**
Course Outcomes	After going through the course, learners will be able to
	1. Explain the basic concepts, history, and importance of Artificial Intelligence.
	2. Identify different types of AI and their working principles.
	3. Understand how AI is used in daily life and various industries.
	4. Analyze ethical, social, and future impacts of Artificial Intelligence.
Module 1(Credit 1): Introduction to Artificial Intelligence	
Learning Outcomes	After learning the module, learners will be able to
	1. Define Artificial Intelligence in simple terms.
	2. Differentiate between human intelligence and machine intelligence.
	3. Identify the evolution and milestones of AI.
Content Outline	➢ What is Artificial Intelligence? ➢ Intelligence: Human vs Machine ➢ History and Evolution of AI 1. Early AI ideas 2. Growth of AI in modern times ➢ Examples of AI in Daily Life • Google search, voice assistants

Module 2(Credit 1): How Artificial Intelligence Works (Basic Concepts)	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand how machines learn from data.
	2. Identify basic AI techniques without technical complexity.
Content Outline	➢ **Data and Information** • Importance of data in AI • Types of data (text, image, audio, video) ➢ **Introduction to Machine Learning** • Learning from examples • Supervised and unsupervised learning (basic idea)
Module 3(Credit 1): Applications of Artificial Intelligence	
Learning Outcomes	After learning the module, learners will be able to
	1. Identify AI applications in various fields.
	2. Understand how AI improves efficiency and decision-making.
	3. Analyze real-life AI use cases.
Content Outline	➢ **AI in Daily Life** • Smart phones and smart devices • Chatbots and virtual assistants ➢ **AI in Education** • Personalized learning • Online assessment systems ➢ **AI in Healthcare** • Disease prediction • Medical imaging

Module 4(Credit 1): Ethics, Challenges, and Future of Artificial Intelligence	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand ethical issues related to AI.
	2. Identify challenges and limitations of AI systems.
Content Outline	➢ **Ethical Issues in AI** • Bias and fairness • Privacy and data security ➢ **Social Impact of AI** • Job automation and employment • Human–AI collaboration

.

Semester - I

.1.4 Vocational Skill Courses (VSC-S1)

Course Title	**Computational Thinking & Problem Solving**
Course Credits	**2 Credits**
Course Outcomes	After learning the module, learners will be able to
	1. Understand the fundamentals of computational thinking.
	2. Apply problem-solving steps to real-world problems.
	3. Develop flowcharts and algorithms to represent solutions.
	4. Use decomposition, pattern recognition, and abstraction.
	5. Translate logical solutions into pseudo-code.
	6. Debug solutions for correctness and efficiency.
Module 1 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Define computational thinking.
	2. Apply the problem-solving cycle.
	3. Identify and analyze problems.
	4. Break down problems into smaller parts.
Content Outline	• **Introduction**: Computational thinking concepts. • **Problem-Solving Steps**: Define, analyze, explore, evaluate solutions. • **Decomposition**: Breaking problems into sub-problems.
Module 2 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Identify patterns in problem-solving.

	2. Use abstraction to focus on relevant details.
	3. Represent problems visually.
	4. Apply logical steps for problem solutions.
Content Outline	• **Pattern Recognition**: Detect similarities in problems. • **Abstraction**: Ignore irrelevant details. • **Representation**: Models, tables, charts.
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: • Identify a real-world problem and write the problem statement. • Decompose a problem into smaller sub-problems. • Brainstorm possible solutions in groups.	
Module 2: • Identify patterns in different problem statements. • Apply abstraction by eliminating irrelevant details. • Represent a problem in table or chart form.	

References:

1. Wing, J. M. (2006). *Computational Thinking*. Communications of the ACM.
2. Selby, C. C. (2015). *Computational Thinking: An Introduction*. Routledge.
3. Lee, V. R. (2014). *Computational Thinking in Practice*. Springer.
4. Kaldewey, T. (2017). *Problem Solving with Algorithms*. Free Online Textbook.
5. BBC Bitesize. (2022). *Computational Thinking Resources*. BBC Education.
6. Pólya, G. (2004). *How to Solve It*. Princeton University Press.

Assessment:

Internal Assessment: (50 marks)

Evaluation Scheme:

Evaluation Parameters	Description / Evaluation Points	Marks
Problem Understanding	Clear statement of problem	10
Algorithm Design	Logical correctness of algorithm	15
Flowchart & Pseudo-code	Correct use of representation	10
Debugging Skills	Identification and correction of errors	5
Class Participation	Active participation in problem-solving exercises	10

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: ATM Simulation Flowchart – Create an algorithm and flowchart to simulate cash withdrawal from an ATM.

Project 2: Student Result Processing – Design a pseudo-code and flowchart to calculate results with grades.

Semester - I

.1.5 B. Skill Enhancement Courses (SEC)

Course Title	**Introduction to Operating System**
Course Credits	**2 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Understand the role, functions, and importance of an Operating System.
	Explain core OS concepts such as process management, memory management, and file systems.
	Describe CPU scheduling and deadlock concepts with real-world examples.
	Analyze how operating systems manage resources efficiently and securely.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Explain the purpose and functions of an Operating System.
	2. Differentiate between various types of Operating Systems.
	3. Understand process concepts and CPU scheduling mechanisms.
	4. Describe basic inter-process communication concepts.
Content Outline	➢ **Introduction to Operating Systems** • Definition and objectives of Operating System • Functions of an Operating System • Evolution of Operating Systems ➢ **Types of Operating Systems** • Batch Operating System • Multiprogramming and Multitasking OS • Time-Sharing OS • Real-Time Operating System • Distributed and Network Operating Systems

	➢ **Process Management** • Process concept and states • Process Control Block (PCB) • CPU Scheduling concepts • Scheduling algorithms: o FCFS o Shortest Job First (SJF) o Priority Scheduling o Round Robin
Module 2 (Credit 1)	**Memory, File Systems & Security**
	After learning the module, learners will be able to
Learning Outcomes	1. Explain memory management techniques used by Operating Systems.
	2. Describe file system organization and access methods.
	3. Understand deadlocks and resource management.
	4. Identify basic OS security and protection mechanisms.
	5. Explain memory management techniques used by Operating Systems.
Content Outline	➢ **Memory Management** • Logical vs Physical memory • Contiguous and non-contiguous memory allocation • Paging and Segmentation (conceptual) • Virtual memory overview ➢ **File System Management** • File concept and file attributes • File access methods • Directory structures

	• File allocation methods ➢ **Deadlocks** • Deadlock concept and conditions • Deadlock prevention and avoidance (overview) ➢ **Security & Protection** • User authentication • Access control • Overview of OS security threats
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: • Draw and explain process state diagrams • Compare CPU scheduling algorithms using examples • Group discussion on types of operating systems • Short quiz on OS fundamentals	
Module 2: • Case study on memory management in modern OS • File system structure comparison (Windows vs Linux) • Deadlock identification using real-life scenarios • Assignment on OS security mechanisms	

Reference books

1. Silberschatz, Galvin, Gagne – *Operating System Concepts*, Wiley
2. Andrew S. Tanenbaum – *Modern Operating Systems*, Pearson
3. William Stallings – *Operating Systems: Internals and Design Principles*
4. D. M. Dhamdhere – *Operating Systems: A Concept-Based Approach*

Internal Assessment – 50 Marks

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Core Functionality & Feature Coverage	Successful implementation of core Operating System concepts such as process management, memory management, file system concepts, scheduling, synchronization, or system utilities with correct and complete functionality.	10
Code Structure & Best Practices	Programs, simulations, or models are well-structured, modular, properly commented, follow naming conventions, and demonstrate correct application of Operating System logic.	10
User Interface & Design Consistency	Interaction (menu-driven programs, simulations, or demonstrations) is user-friendly, clear, intuitive, and maintains consistent input/output formatting.	10
Input Validation & Exception Handling	Validates all user inputs properly and handles errors or exceptional cases gracefully using appropriate control structures or logical checks.	10
Technical Documentation	Project report is professionally prepared with a clear structure including introduction, problem definition, system design/flow, implementation details, outputs/screenshots, and explanation of results.	10

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: CPU Scheduling Algorithms

- Study and simulate CPU scheduling algorithms: FCFS, SJF, Priority, Round Robin
- Compare Waiting Time and Turnaround Time
- Draw simple Gantt charts
- Explain which algorithm is better and why

Project 2: Memory Management Techniques

- Study memory management in OS
- Explain Fixed & Variable Partitioning
- Conceptual study of Paging and Segmentation
- Show memory allocation using diagrams/tables
- Discuss fragmentation issues

Semester - I

.1.5 C. Skill Enhancement Courses (SEC)

Course Title	**GitHub & Version Control**
Course Credits	**2**
Course Outcomes	After Completion of this Course, students will be able to
	1. Understand the basics of version control systems.
	2. Use Git for local repository creation and management.
	3. Apply branching and merging techniques.
	4. Collaborate on projects using GitHub.
	5. Resolve conflicts and manage pull requests.
	6. Follow best practices in commits and project organization.
Module1 (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Define version control and its importance.
	2. Differentiate between local and remote repositories.
	3. Set up Git on their system.
	4. Create and initialize repositories.
Content Outline	• **Introduction to Version Control**: Centralized vs. Distributed VCS. • **Git Basics**: Installation, configuration. • **Repository Creation**: git init, git clone.
Module2 (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Commit changes to Git repository.
	2. Track file changes and versions.
	3. Use staging area effectively.

	4. Understand commit history.
Content Outline	• **Staging & Committing**: git add, git commit. • **Tracking Changes**: git status, git log. • **Commit Best Practices**: Writing commit messages.

Reference Books

1. Loeliger, J., & McCullough, M. (2012). *Version Control with Git* (2nd ed.). O'Reilly.
2. Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress.
3. Ryder, R. (2020). *Git for Teams*. O'Reilly.
4. Premasiri, R. (2022). *Mastering GitHub*. Packt.
5. Silverman, R. (2020). *Git Essentials*. Packt.
6. Kamei, Y. (2021). *Practical Git and GitHub*. Apress.

Practical Assignments

The Rubric will have the following

Evaluation Parameter	**Description**	**Marks**
Git Setup & Repo Creation	Correct installation and initialization	10
Commit History	Clear, logical commit messages	10
Branching & Merging	Correct use of branches & merges	10
Conflict Resolution	Successful resolution of conflicts	10
Collaboration	Push, pull requests, issue management	10

<u>Assessment:</u>

External Assessment – 50 Marks

End Semester examination of 50 marks for 2 hours' duration will be conducted.

Semester - II

.2.1 Major (Core)

Course Title	**Python Programming**
Course Credit	**4 Credits**
Course Outcomes	1. Understand Python syntax, semantics, and structure.
	2. Write Python programs using control flow and modular programming.
	3. Implement basic data structures like lists, tuples, and dictionaries.
	4. Perform string and file operations.
	5. Develop functions and reusable modules in Python.
Module 1 (Credit	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Set up a Python development environment.
	2. Write and execute basic Python programs.
	3. Understand variables, constants, and data types.
	4. Use basic input/output functions.
Content Outline	➢ Introduction **to Python**: Features, IDEs (IDLE, Jupyter Notebook, PyCharm). • Syntax: Indentation, variables, constants. • Data **Types**: int, float, string, boolean. • **Input/Output**: input(), print() functions. • Operators: Arithmetic, relational, logical, assignment, precedence.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply conditional statements in Python programs.
	2. Use loops for iterative operations.

	3. Write programs using nested decision-making and loops.
	4. Apply control flow to solve problems.
Content Outline	• Decision Control: if, if-else, nested if. • Loops: for, while, nested loops. • Control Statements: break, continue, pass. • Practical Exercises: Number operations, table generation, factorial, Fibonacci sequence.
Module 3 (Credit 1)	
Learning Outcomes	**After learning this module learners will be able to understand**
	1. Use built-in data structures effectively
	2. Perform file operations in Python
Content Outline	• Lists and List Operations • Tuples and Sets • Dictionaries and Dictionary Operations • String Manipulation Techniques • File Handling: open, read, write, append • Working with CSV and Text Files
Module 4 (Credit 1)	
Learning Outcomes	**After learning this module learners will be able to**
	1. Implement object-oriented concepts in Python
	2. Handle runtime errors using exception handling
Content Outline	➢ **Introduction to Object-Oriented Programming** • Classes and Objects • Constructors and Methods ➢ **Inheritance and Polymorphism** • Encapsulation and Abstraction • Exception Handling: try, except, finally • User-defined Exceptions

Activities to be done in the class towards Comprehensive Continuous Evaluation
Module 1: • Write a Python program to swap two numbers. • Demonstrate different data types with examples. • Evaluate expressions with different operators and precedence.
Module 2: • Write a program to find the factorial of a number using loops. • Create a program to print multiplication tables. • Implement a program to check prime numbers in a range.
Module 3: • Write a program to perform insertion, deletion, and update operations on a list. • Write a program to remove duplicate elements from a list. • Write a program to perform union, intersection, and difference operations on sets. • Write a program to create a dictionary of student marks and calculate total and average. • Write a program to count the frequency of characters in a given string using a dictionary. • Write a program to read a text file and count the number of lines, words, and characters.
Module 4: • Write a program to create a class and object to display student details. • Write a program demonstrating the use of constructors and instance methods. • Write a program to demonstrate single or multilevel inheritance. • Write a program to demonstrate method overriding (polymorphism). • Write a program to implement encapsulation using private variables. • Write a program to handle divide-by-zero and file-not-found exceptions

Reference Books:

1. Korth, H. F., & Silberschatz, A. (2010). Database system concepts (6th ed.). McGraw- Hill.
2. Elmasri, R., & Navathe, S. B. (2010). Fundamentals of database systems (6th ed.). McGraw-Hill.
3. Bayross, P. (n.d.). Oracle: The complete reference. BPB Publications.
4. Datapro InfoWorld Ltd. (n.d.). Upgrade to Oracle 8.
5. Widom, J., & Wiederhold, G. (1995). Database design. McGraw-Hill.

Assessment:

Internal Assessment: (Marks 50) Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below:

Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Clear understanding of Python fundamentals, data structures, OOP concepts, and file handling.Correct explanation of concepts such as lists, dictionaries, classes, inheritance, and exceptions.	**10**
Accuracy and Completeness	All assigned programs are attempted.Programs are syntactically correct and produce accurate output.Proper use of Python statements, functions, and file operations.	**15**
Application and Analysis	Correct application of Python concepts to solve problems.Uses appropriate data structures, OOP principles, and exception handling.Demonstrates logical thinking and problem analysis.	**10**
Presentation and Neatness	Code is well-formatted and readable.Proper indentation, meaningful variable names, and comments.Organized presentation of programs and outputs.	**5**
Innovation and Effort	Uses real-world problem scenarios (e.g., student management, banking, library systems).Attempts additional features, validations, or enhancements beyond basic requirements.	**5**
Timely Submission	Programs and records are submitted within the stipulated deadline.Late submissions may attract a deduction of up to 5 marks unless exempted.	**5**
Total		**50 Marks**

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – II

.2.2 Major (Core)

Course Title	**Discrete Mathematics for AI**
Course Credits	**2 Credits**
Course Outcomes	1. Understand the foundational concepts of sets, relations, and functions used in AI.
	2. Apply propositional logic, Boolean algebra, and proof techniques to reasoning tasks.
	3. Solve combinatorics and probability problems relevant to AI models.
	4. Represent and analyze AI problems using graph theory.
	5. Relate discrete mathematical concepts to AI algorithms such as search, classification, and optimization.
	6. Develop logical reasoning skills to support AI decision-making.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Define and represent sets and operations.
	2. Analyze relations and determine their properties.
	3. Apply functions and mappings to AI problem spaces.
Content Outline	• **Sets**: Definition, notations, operations (union, intersection, complement, Cartesian product). • **Relations**: Types, properties (reflexive, symmetric, transitive), equivalence relations. • **Functions**: Domain, range, types (one-to-one, onto, bijection), composition, inverse functions. • **AI Application**: Mapping input features to outputs in AI models.

<table>
<tr><td colspan="2">Module 2 (Credit 1)</td></tr>
<tr><td rowspan="4">Learning Outcomes</td><td>After learning the module, learners will be able to</td></tr>
<tr><td>1. Apply propositional and predicate logic to formal reasoning.</td></tr>
<tr><td>2. Construct truth tables and derive valid conclusions.</td></tr>
<tr><td>3. Simplify Boolean expressions for AI hardware implementation.</td></tr>
<tr><td>Content Outline</td><td>• Propositional Logic: Logical connectives, truth tables, tautologies, contradictions.
• Predicate Logic: Quantifiers, translating English sentences to logic.
• Boolean Algebra: Laws, simplification, Karnaugh maps.
• AI Application: Logical reasoning in expert systems, circuit design for AI accelerators.</td></tr>
<tr><td colspan="2">Activities to be done in the class towards Comprehensive Continuous Evaluation</td></tr>
<tr><td colspan="2">Module 1:
• Represent real-world dataset categories using Venn diagrams.
• Create equivalence relation examples in AI dataset grouping.
• Apply functions to model feature-to-label mappings.</td></tr>
<tr><td colspan="2">Module 2:
• Build truth tables for AI decision-making rules.
• Translate AI classification statements into predicate logic.
• Simplify Boolean expressions for circuit representation of AI logic gates.</td></tr>
</table>

Reference Books

1. Rosen, K. H. (2019). *Discrete Mathematics and Its Applications*. McGraw-Hill.
2. Tremblay, J. P. (2018). *Discrete Mathematical Structures with Applications to Computer Science*. McGraw-Hill.
3. Lipschutz, S., & Lipson, M. (2017). *Discrete Mathematics*. McGraw-Hill.

4. Epp, S. S. (2011). *Discrete Mathematics with Applications*. Cengage.

Assessment:

External Assessment – 50 Marks

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester - II

.2.3 Vocational Skill Courses (VSC-S2)

Course Title	**Introduction to Artificial Intelligence**
Course Credits	**2 Credits**
Course Outcomes	1. Understand the foundational concepts, history, and evolution of AI.
	2. Differentiate between various branches and types of AI: narrow, general, and superintelligence.
	3. Explore core areas such as search algorithms, knowledge representation, and expert systems.
	4. Explain the basics of machine learning, neural networks, and natural language processing.
	5. Analyze AI's role in real-world domains like healthcare, robotics, education, and finance.
	6. Identify challenges and ethical concerns in AI implementation
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Define Artificial Intelligence and its subfields.
	2. Explain the historical milestones and evolution of AI.
	3. Describe types of AI: narrow, general, and strong AI.
	4. Differentiate between human and machine intelligence.

<table>
<tr><td>Content Outline</td><td>

- **Concept:**
  Definition and Goals of AI, Characteristics of AI, Foundations of AI (philosophy, logic, mathematics, psychology, computer science), Applications of AI in various domains.

- **History of AI:**
  Timeline and Milestones in AI (1950–present), Turing Test, Dartmouth Conference, early AI programs (ELIZA, SHRDLU), AI winters and revivals.

- **Types of AI:**
  Narrow AI, General AI, Super AI. Reactive Machines, Limited Memory, Theory of Mind, Self-aware AI.

- **Basic Terminology:**
  Intelligent Agent, Environment, Rationality, Perception-Action Cycle.

</td></tr>
<tr><td colspan="2">Module 2 (Credit 1)</td></tr>
<tr><td rowspan="5">Learning Outcomes</td><td>After learning the module, learners will be able to</td></tr>
<tr><td>1. Define agents and environments in AI.</td></tr>
<tr><td>2. Classify different types of intelligent agents.</td></tr>
<tr><td>3. Apply goal formulation and problem-solving strategies.</td></tr>
<tr><td>4. Demonstrate uninformed and informed search techniques.</td></tr>
<tr><td>Content Outline</td><td>

- **Concept**:
  Definition of Agents, Types of Agents: Simple Reflex, Model-Based, Goal-Based, Utility-Based, Learning Agents.

- **PEAS Framework**:
  Performance Measure, Environment, Actuators, Sensors.

- **Problem-Solving Techniques**:
  State-Space Representation, Formulation of Problems, Problem Types (Toy vs. Real-world).

- **Search Strategies**:
  Uninformed Search (BFS, DFS), Informed Search (Greedy, A*), Heuristic Functions.

</td></tr>
</table>

Activities to be done in the class towards Comprehensive Continuous Evaluation

Module 1:

1. Create a concept map of AI including definitions, goals, subfields, and types (Narrow AI, General AI, Super AI).
2. Design a timeline poster or slide deck showcasing major milestones in AI history (e.g., 1950 – Turing Test, 1997 – Deep Blue).
3. Differentiate between types of AI through a comparison table (Reactive, Limited Memory, Theory of Mind, Self-aware).
4. Prepare flashcards for terminology such as intelligence, agent, rationality, and environment.
5. Research & present on one real-world application of AI (e.g., AI in agriculture, robotics, or virtual assistants).
6. Write a short essay explaining the difference between human and machine intelligence.
7. Conduct a classroom quiz using Kahoot/Google Forms on AI basics and history.
8. Group discussion on myths about AI (e.g., AI will take over the world) and misconceptions in media.
9. Watch a short AI documentary clip (e.g., from YouTube/Crash Course) and summarize its key takeaways.
10. Demonstrate identification of AI features in tools/apps students use daily (e.g., recommendations on YouTube, voice assistants).

Module 2:

1. Design a rational agent for a smart environment (e.g., smart room, self-driving car) and define its PEAS specification.

2. Simulate an agent-environment interaction using role-play in class (student as agent, others as environment elements).

3. Draw a flowchart to represent the decision-making process of a goal-based agent.

4. Implement a Python program for Breadth-First Search (BFS) on a grid or tree structure.

5. Compare search strategies using a real-world problem (e.g., pathfinding, route planning) and represent them visually.

6. Create a state-space representation for a toy problem (e.g., 8-puzzle, water jug).

7. Analyze the efficiency of uninformed vs. informed search using sample problems.

8. Group activity: Design a search problem and ask peers to solve it using A* algorithm manually.

9. Mini quiz on types of intelligent agents and search strategies.

10. Case-based discussion on real-world AI agent applications (e.g., GPS, chess-playing AI).

Reference Books

1. Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson Education.

2. Rich, E., Knight, K., & Nair, S. B. (2017). Artificial Intelligence (3rd ed.). Tata McGraw-Hill.

3. Ertel, W. (2018). Introduction to Artificial Intelligence (2nd ed.). Springer.

4. Nilsson, N. J. (2010). The Quest for Artificial Intelligence: A History of Ideas and Achievements. Cambridge University Press.

5. Poole, D., & Mackworth, A. (2017). Artificial Intelligence: Foundations of Computational Agents (2nd ed.). Cambridge University Press.

6. Kurzweil, R. (1990). The Age of Intelligent Machines. MIT Press.

Assessment:

Internal Assessment – 50 Marks

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: Using Python, create a rule-based expert system that provides basic medical advice based on symptoms input by users. The system should apply IF-THEN rules and explain its decision using simple logic.

Project 2: Build an AI-powered pathfinding tool that uses the A* search algorithm to find the shortest path in a 2D grid (e.g., for a game map or navigation system). Visually demonstrate the algorithm's steps and heuristic-based choices.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Code Implementation	● Accuracy in writing and executing the C program, correct logic, syntax	15
Problem Understanding	● Clarity of problem interpretation and correct approach	10
Output Correctness	● Program compiles without errors and gives correct output	10
Code Quality & Structure	● Proper use of indentation, comments, meaningful variable names, modularity	5
Debugging Skills	● Ability to identify and fix logical/syntax errors during execution	5
Lab Record/Journal	● Completeness, correctness, formatting, and timely submission	5

Semester - II

.2.4. Vocational Skill Courses (VSC-S3) (With Respect to Minor)

Course Title	**Computer Organization and Architecture**
Course Credits	**2 Credits**
Course Outcome	**After Completion of this Course, students will be able to**
	1. Apply basic concepts to understand the structure of computers.
	2. Analyze the working of different interrupts and mapping techniques, and study register organization.
	3. Evaluate different addressing modes and their functionalities.
	4. Design and demonstrate the working of the central processing unit, including RISC and CISC architectures.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge to understand the basic structure of a computer.
	2. Analyze memory organization and its components.
	3. Evaluate different memory types and their roles within a computer system.
	4. Design and implement memory organization techniques in computing scenarios.
Content Outline	• **Basic Structure of computers:** Basic organization of computer, Intel 8086 Architecture, Basic Measures of Computer Performance, CPU: Registers, Computer Function: Instruction Cycle, Interrupts, Interconnection Structures, Bus Interconnection, Peripheral Component Interconnection (PCI). • **Memory Organization:** Classifications of primary and secondary memories. Types of RAM (SRAM, DRAM, SDRAM, DDR, SSD) and ROM, Characteristics of memory, Memory hierarchy: cost and performance measurement.

	• **Cache Memory:** Principles, Elements of cache design (Size, Mapping, Replacement, Write policies, Block size) Virtual Memory Concept.)
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply various I/O techniques, including programmed I/O, interrupt-driven I/O, and direct memory access, to manage external devices and I/O modules.
	2. Analyze the instruction sets, operand types, operations on operands, and addressing modes of the 8086 processor.
	3. Evaluate the characteristics and performance differences
	4. between RISC and CISC architectures, including their pipelining capabilities.
	5. Design processor and register organizations based on the principles of RISC and CISC instruction execution.
Content Outline	• **Input/Output:** External devices, I/O Modules, Programmed I/O, Interrupted-Driven I/O, Direct Memory Access. • **Central Processing Unit:** Instruction sets: instruction characteristics, Types of operands, Types of operations on operands, addressing modes of 8086 processor, Processor Organization, Register organization. • **RISC:** Instruction Execution, RISC Characteristics, and RISC Pipelining, RISC Vs. CISC, Reduced Instruction Set Computers (RISCs), Introduction to CISC. CISC Characteristics.
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: 1. Study the basic organization of a computer system, focusing on Intel 8086 Architecture. 2. Explore the fundamental measures of computer performance and the role of CPU registers in the instruction cycle. 3. Classify primary and secondary memories, and analyze their classifications and characteristics. 4. Evaluate different types of RAM (SRAM, DRAM, SDRAM, DDR) and ROM,	

considering their cost and performance metrics.
5. Study the principles of cache memory.
6. Design elements of cache memory, including size, mapping, replacement policies, write policies, and block size considerations.
7. Discuss the concept of virtual memory and its implementation in modern computing.
8. Design a memory hierarchy system that optimizes both cost and performance for a specific computing scenario.
9. Implement the designed memory organization techniques in a simulation environment using appropriate tools or programming languages.

Module 2:

1. Implement various I/O techniques such as programmed I/O, interrupt-driven I/O, and direct memory access (DMA) to manage external devices and I/O modules.
2. Analyze the effectiveness and suitability of each technique in different computing scenarios.
3. Analyze the instruction sets, including characteristics, types of operands, and operations on operands of the Intel 8086 processor.
4. Evaluate the addressing modes and their implications for program execution efficiency.
5. Compare and contrast RISC and CISC architectures in terms of instruction execution efficiency and complexity.
6. Evaluate the advantages and disadvantages of RISC pipelining and its impact on processor performance.
7. Design a processor architecture based on either RISC or CISC principles, considering instruction set design and register organization.
8. Discuss how the designed architecture enhances performance and efficiency in specific computing tasks.

References book:

1. **Computer Organization and Architecture** — *P. N. Basu* (Vikas Publishing) — concise and aligned with many Indian university syllabi.
2. **Computer Organization and Architecture** — *Atul P. Godse* (Technical Publications) — covers components, instructions, I/O and performance.
3. **Computer Organization** — *G. V. Anjaneyulu* (Himalaya Publishing House) — details basics of architecture and I/O.

Assessment:

Internal Assessment: (50 marks) Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Code Implementation	Correct syntax, structure, and execution	15
Problem Solving Approach	Logical problem-solving methods applied	10
Output Correctness	Program produces expected output	10
Code Quality	Indentation, variable names, comments	5
Debugging Skills	Ability to fix syntax/logical errors	5
Lab Record/Assignments	Completeness and correctness	5

Semester - II

.2.6 Skill Enhancement Courses (SEC)

Course Title	**Linux Operating System**
Course Credits	**2**
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply Linux environment skills effectively for various tasks and operations.
	2. Analyse and utilize basic Linux commands and shell scripting techniques.
	3. Evaluate file system creation, directory management, and related operations using Linux programs.
	4. Design and implement solutions using a diverse set of standard Linux programming and development tools.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply basic Linux commands proficiently for various tasks and operations.
	2. Analyses the significance of Linux architecture and its key features.
	3. Evaluate Linux utilities for creating and managing simple file processing operations effectively.
	4. Design and demonstrate memory management techniques in file handling, including file and region locking mechanisms.
Content Outline	• **Introduction to Linux** Introduction to Unix architecture, General-purpose utilities, All basic command's introduction, Usage of commands. • **Files and Directories:** File Concept, File types, File System Structure, File metadata- Inodes, kernel support for files, file System calls for file I/O operations- open, create, read, write, close. • **Directories-** mkdir, rmdir, chdir, obtaining current

	working directorygetcwd, directory contents, scanning directories opendir, readdir, closedir, rewind dir functions.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply the concept of filters in Linux to manipulate data effectively.
	2. Analyse different editors available in Linux and their functionalities.
	3. Design and implement shell scripts capable of performing complex tasks in a shell programming environment.
	4. Evaluate the role and functionalities of the Linux kernel in system operations.
Content Outline	• **The vi Editor** • Introduction to the vi editor of Linux. all commands. • **Introduction to filters** • Simple Filters, Filters using regular expressions - use of grep command • **Introduction to shell concept and writing shell script-** What is kernel, What is Shell, what is terminal, Advantage and disadvantage, first simple Shell Programme.

Activities to be done in the class towards Comprehensive Continuous Evaluation

Module 1:

1. Basic Linux Commands Proficiency:
   - Practice using a range of basic Linux commands for file and directory management, navigation, and system operations.
   - Apply these commands to perform tasks such as file creation, modification, deletion, and directory navigation

2. Analysis of Linux Architecture:
   - Analyze the architecture of Unix/Linux systems, including its components and functionalities. Evaluate the significance of key features like multi-user capability, security models, and file system structure in Unix/Linux.

3. Evaluation of Linux Utilities:
   - Evaluate Linux utilities for file processing operations, including creation, deletion, copying, and searching files and directories.
   - Analyze the efficiency and effectiveness of these utilities in real-world scenarios.

4. Design and Demonstrate Memory Management Techniques:
   - Design a system for managing file memory, including techniques like file and region locking mechanisms.
   - Implement and demonstrate these techniques using Linux system calls and utilities.

Module 2:

1. Application of Filters:
   - Apply filters in Linux to manipulate data effectively, including simple filters and those using regular expressions.
   - Use commands like grep for text pattern searching and filtering.

2. Analysis of Linux Editors:
   - Analyze different editors available in Linux, focusing on the vi editor.
   - Explore and practice all essential commands and functionalities of the vi editor for text editing and manipulation.

3. Design and Implementation of Shell Scripts:
   - Design shell scripts capable of performing complex tasks in a shell programming environment. Develop scripts that automate file management, data processing, or system administration tasks.

4. Evaluation of Linux Kernel Functionalities:
   - Evaluate the role and functionalities of the Linux kernel in system operations, including process management, memory management, and device management. Discuss the advantages and disadvantages of kernel-level operations and their impact on system performance.

Reference books:

1. Card, R., Dumas, E., & Mevel, F. (2003). The Linux kernel book. Wiley.
2. Das, S. (2017). Unix concepts and applications (4th ed.). TMH.
3. Suehring, S. (2002). MySQL Bible. John Wiley & Sons.
4. Lerdorf, R., & Tatroe, K. (2002). Programming PHP. O'Reilly Publications. 5. Collings, T., & Wall, K. (Wiley). (2003). Red Hat Linux network and system administration (3rd ed.).
5. Mathews, N. (2007). Beginning Linux programming (4th ed.). Wrox Press.
6. Koparkar, P. (2001). Unix for you. Tata McGraw-Hill.
7. Kanetkar, Y. P. (2013). Unix shell programming (1st ed.). BPB Publications.

Assessment

Internal Assessment – 50 Marks

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Core Functionality & Feature Coverage	All required Linux features and functions are implemented (file ops, filters, memory mgmt, shell scripts, vi editor, etc.) with full Functionality	10
Code Structure & Best Practices	Code is modular, well-commented, follows naming conventions, and uses appropriate Linux commands and logic	10
User Interface & Design Consistency	Command-line interaction is user-friendly, intuitive, with helpful prompts and consistent Format	10
Input Validation & Exception Handling	Validates all user inputs (e.g., file names, paths, user options); handles all exceptions gracefully using built-in checks or control structures	10
Technical Documentation	Professional report with clear structure: introduction, architecture, code snippets, output screenshots, and detailed explanation	10

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: Linux Automation & Memory Safety Toolkit

A multi-functional script that automates system tasks (file handling, process monitoring, logs) and implements memory/file locking mechanisms.

Project 2: Linux Command Trainer with vi & Filters

Build an interactive command-line learning environment for beginners. Create interactive shell script tutorials that teach use of:
grep, awk, sed, cut, sort, etc. Regular expressions and filters
Build an in-terminal vi editor tutorial, explaining modes, commands, macros
Write a formal report, use screenshots, and prepare a demo presentation with voice-over

Course Syllabus

Semester - III

.3.1 Major (Core)

Course Title	**Numarical Method**
Course Credits	**4 Credits**
Course Outcomes	**After Completion of this Course, students will be able to**
	1. To Study error estimation in numerical methods
	2. To study different methods of numerical methods
	3. To enable the students to use nonlinear and linear methods to analyze and understand problems in the same
	4. To demonstrate how the numerical methods help to develop thinking ability
	5. To demonstrate how the interpolation helps to develop thinking ability
Module 1 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply knowledge to understand the basic structure of a computer.
	2. Analyze memory organization and its components.
	3. Evaluate different memory types and their roles within a computer system.
	4. Design and implement memory organization techniques in computing scenarios.
Content Outline	• Floating Point Arithmetic and Errors: Floating Point Representation, Sources of Errors. Propagated Errors

Module 2 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	1. To Understand the methods of linear equation
Content Outline	• Solution to linear equation: ○ Matrix inversion method, Gauss Elimination method.
Module 3 (Credit 1)	
	After learning the module, learners will be able to
Learning Outcomes	1. To Understand the methods of non linear equation
	2. Apply DMA controller concepts to improve data transfer efficiency.
	3. Evaluate how instruction formats and addressing modes affect system performance
Content Outline	• Solution to nonlinear equation : Bisection method , Newton Raphson Method, Regula Falsi method
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. To understand basics of interpolation
Content Outline	Interpolation: Operator Δ ∇, E and their relations. Newton's Interpolation Formula (Forward and Backward). Lagrange's Interpolation Formula. Newtons divided difference formula.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE):

Module 1:

1. Solve a set of problems related to error estimation and propagated errors.
2. Solve examples of linear equations using the Matrix inversion and Gauss Elimination method.
3. Solve a set of problems using the Bisection method and Newton-Raphson method.

Module 2:

1. Solve numerical problems using Newton's Forward and Backward Interpolation

methods.
2. Solve examples to interpolate values using Lagrange's interpolation formula.

Module 3:

1. Solve problems using numerical methods and compare results with analytical solutions.
2. 2.Solve examples using the Trapezoidal and Simpson's rules.

Module 4:

1. Solve numerical differentiation problems using first and second-order derivatives.

Reference Books:

1. S.S. Sastry; Introductory Methods of Numerical Analysis, 3rd edition, Prentice Hall of India.
2. H.C. Saxena; Finite differences and Numerical Analysis, S. Chand and Company.
3. JAIN M K_NUMERICAL METHODS FOR 8ED.(699)
4. Numerical Methods for Scientists and Engineers Richard Hamming
5. A First Course in Numerical Methods Uri M. Ascher Chen Greif

Assessment:

Internal Assessment: (Marks 50)

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Demonstrates deep understanding of core concepts with insightful analysis.	10
Technical Accuracy	All technical content is accurate and complete; terminology is used correctly.	15
Practical Application	Concepts are effectively applied in real-world or simulated scenarios.	10
Presentation & Documentation	Information is well-organized; formatting, visuals, and documentation are clear.	5

Creativity and Design Skills	Demonstrates innovation, originality, and thoughtful design.	5
Timely Submission	Submitted before or on time with all required elements.	5

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: Write a program to demonstrate linear solutions methods.

Project 2: Find real world the applications of interpolation

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester - III

.3.2 Major (Core)

Course Title	**Data Structure & Algorithms**
Course Credits	**4 credits**
Course Outcomes	1. Apply abstract data structures using arrays and linked lists.
	2. Analyze the efficiency and performance of algorithms.
	3. Evaluate data structures like arrays, linked lists, stacks, queues, binary trees, and graphs for different applications.
	4. Design and apply appropriate searching and sorting techniques.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply various operations on linear data structures such as arrays, linked lists, stacks, and queues.
	2. Analyze algorithm characteristics, including space and time complexity, using asymptotic notation.
	3. Evaluate the advantages and disadvantages of different types of linked lists and their operations.
	4. Design and implement stack and queue structures, and apply their operations in different contexts.

Content Outline	• **Algorithm Analysis:** Algorithm Characteristics, Space complexity, Time complexity. Asymptotic notation (Big O, 0, Omega and Theta) • **Introduction:** Introduction to data structure, Classification of data structure, Operations performed on data structures, Linear data structure, arrays, operations on an array. • **Linked List:** Introduction, Key terms, Advantages & disadvantages, Linear linked lists - Types (Singly, Doubly, Circular) Operations (Inserting, Deleting nodes) • **Stack:** Introduction, Stack implementation, Operations on stack (Push Pop), Implementation of stack using pointer, Applications of stack, Infix prefix, postfix notations • **Queue:** Introduction and Queue implementation, Operations.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply various traversal techniques to binary trees and graphs.
	2. Analyze different sorting algorithms such as Bubble Sort, Quick Sort, and Heap Sort.
	3. Evaluate the efficiency of search algorithms like Linear Search 4. and Binary Search.
	5. Design binary trees and graphs using array and linked list representations.

<table>
<tr><td>Content Outline</td><td>

- **Trees:**
  Introduction, terminology, Binary tree, Strictly Binary tree, Complete Binary tree, Binary tree representation as Array and Linked lists, Traversal (Inorder, Preorder,Post order), Binary Search Tree

- **Graphs:**
  Introduction, terminology, Graph representation, Applications of graph, Graph traversal (BFS, DFS, Shortest

- **Sorting & Searching:**
  Bubble Sort, Selection Sort, Quick Sort, Heap Sort, Insertion Sort. Linear Search, Binary Search

</td></tr>
<tr><td colspan="2">Module 3 (Credit 1) Data Structure Practical Approach</td></tr>
<tr><td rowspan="5">Learning Outcomes</td><td>After learning the module, learners will be able to</td></tr>
<tr><td>1. Apply various operations on linear data structures such as arrays, linked lists, stacks, and queues.</td></tr>
<tr><td>2. Analyze algorithm characteristics, including space and time complexity, using asymptotic notation.</td></tr>
<tr><td>3. Evaluate the advantages and disadvantages of different types of linked lists and their operations.</td></tr>
<tr><td>4. Design and implement stack and queue structures, and apply their operations in different contexts.</td></tr>
<tr><td>Content Outline</td><td>

- **Arrays:**
  Implementations of Array and Operations- Insertion, deletion of an element from one dimensional array, Traversing of array

- **Linked List:**
  Implementation of List and Linked List and Operations- Inserting, Deleting of nodes etc

- **Stack:**
  Stack Implementation, Operations on stack (Push Pop). Implementation of stack

- **Queue:**
  Implementation of Queue Implementation,Operations on queue(Insertion and deletion)

</td></tr>
</table>

Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply various traversal techniques to binary trees and graphs.
	2. Analyze different sorting algorithms such as Bubble Sort, Quick Sort, and Heap Sort.
	3. Evaluate the efficiency of search algorithms like Linear Search and Binary Search.
	4. Design binary trees and graphs using array and linked list representations.
Content Outline	• **Trees** Implementation of tree as Array and Linked lists and Traversal (Inorder, Preorder, Postorder) • **Graphs** Implementation of Graph traversal (BFS, DFS) • **Sorting and Searching** Implementation of searching (Sequential, Binary search) Sorting (Bubble sort, Selection sort, Insertion Sort.)

Activities to be done in the class towards Comprehensive Continuous Evaluation

Module 1: Implementing and Analyzing Linear Data Structures

Implement the following linear data structures using your preferred programming language:

Array

Singly Linked
List Doubly
Linked List
Circular Linked
List Stack
Queue

Write functions for basic operations on these data structures:

For arrays: insertion, deletion, and traversal.
For linked lists: insertion, deletion, and
traversal. For stacks: push and pop.
For queues: enqueue and dequeue.

Algorithm Analysis:

Analyze the time complexity of the implemented operations using asymptotic notation (Big O, Omega, Theta).

Module 2: Traversal and Sorting Techniques

1. **To apply traversal techniques to binary trees and graphs, analyze sorting algorithms, evaluate search algorithms, and design tree and graph structures.**
2. **Implement the following types of binary trees:**

   Binary Tree
   Strictly Binary Tree
   Complete Binary Tree

   Represent these trees using arrays and linked lists.
3. **Implement the traversal methods for binary trees:**

   Inorder Preorder Postorder
4. **Implement graph representations using adjacency matrix and adjacency list. Implement graph traversal algorithms:**

   Breadth-First Search (BFS) Depth-First Search (DFS)
5. **Sorting Algorithms:Implement the following sorting algorithms:**

   Bubble Sort Selection Sort Quick Sort Heap Sort Insertion Sort
6. **Search Algorithms:Implement Linear Search and Binary Search.**

Module 3:

Implementing and Analyzing Linear Data Structures

1. **Implement the following linear data structures using your preferred programming language:**

   Array

   Singly Linked
   List Doubly
   Linked List
   Circular Linked
   List Stack
   Queue

2. **Write functions for basic operations on these data structures:**

   For arrays: insertion, deletion, and traversal.
   For linked lists: insertion, deletion, and
   traversal. For stacks: push and pop.
   For queues: enqueue and dequeue.

3. **Algorithm Analysis:**

Analyze the time complexity of the implemented operations using asymptotic notation (Big O, Omega, Theta).

Module 4

Traversal and Sorting Techniques

1. **To apply traversal techniques to binary trees and graphs, analyze sorting algorithms, evaluate search algorithms, and design tree and graph structures.**
2. **Implement the following types of binary trees:**

   Binary Tree
   Strictly Binary Tree
   Complete Binary Tree

   Represent these trees using arrays and linked lists.
3. **Implement the traversal methods for binary trees:**

   Inorder Preorder Postorder
4. **Implement graph representations using adjacency matrix and adjacency list. Implement graph traversal algorithms:**

   Breadth-First Search (BFS) Depth-First Search (DFS)
5. **Sorting Algorithms:Implement the following sorting algorithms:**

   Bubble Sort Selection Sort Quick Sort Heap Sort Insertion Sort
6. **Search Algorithms:Implement Linear Search and Binary Search.**

Reference Books

1. Trembley, P., & Sorrenson, R. (2005). Data Structure. McGraw Hill Publication. Lipschuiz, S. (2014). Data structures (Schaum's Outline Series). McGraw Hill Publication. Horowitz, E., & Sawhney, S. (2008). Fundamentals of Computer Algorithms. Computer Sci.P
2. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1982). Data Structures and Algorithms. Pearson
3. Abhyankar, A. (2016) Data Structures and Files. Nirali Prakashan, Educational Publishers Baluja, G. S. (2022). Data Structures Through C. VISIONIAS
4. Loomis, M. E. S. (1989). Data Management and File Structures (2nd ed.). Prentice Hall. Samanta, D. (2009). Classical Data Structures. PHI, New Delhi.

Assessment:

Internal Assessment – 50 Marks

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Ability to define, describe, and explain core data structures concepts clearly and accurately.	20
Practical Implementation	Competence in coding data structure operations in C, with focus on correctness, memory handling, and efficiency.	30
Problem Solving & Analysis	Ability to choose appropriate data structures, analyze complexity, and solve given problems effectively.	20
Design & Application	Skill in applying data structures to real-world problems (e.g., graphs, trees, queues), and implementing custom solutions.	10
Timely Submission	Submitted within the deadline. Late submission deducts up to 5 marks unless exempted.	20

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1:

Using C language create a library management system that handles book borrowing, user registrations, book searches, and reservations
(Use: Hash Tables for fast lookups of books, Linked Lists for maintaining queues of borrowed books, Trees for maintaining ordered catalogues)

Project 2:

Using C language create a flight reservation system where users can book, cancel, and search for flights. (Use: Arrays or Lists for flight schedules, Trees for efficient searching of available flights, Queues for handling waiting lists)

External Assessment – 50 Marks

End Semester examination of 50 marks for 2 hours' duration will be conducted

Semester - III

.3.3 Major (Core)

Course Title	**Introduction to Data Science**
Course Credit	**2 Credits**
Course Outcomes	After successful completion of the course, learners will be able to: 1. Understand fundamental concepts of Data Science, data types, and the data science lifecycle. 2. Identify different data sources and real-world applications of Data Science. 3. Analyze datasets using data preprocessing and basic statistical techniques. 4. Interpret data insights to support data-driven decision-making.
Module 1 (Credit 1) Introduction to Data Science	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand the core concepts and scope of Data Science.
	2. Identify different types of data and their sources.
	3. Explain the Data Science lifecycle and its importance.
	4. Analyze real-world applications of Data Science across domains.
Content Outline	**Introduction to Data Science** • Definition and scope of Data Science • Difference between Data Science, Data Analytics, and Big Data • Data Science lifecycle: data collection, data preparation, analysis, interpretation, and decision-making • Types of data: structured, semi-structured, and unstructured • Sources of data: databases, web data, sensors, surveys • Role of Data Science in organizations

	Applications of Data Science • Data Science in Healthcare • Data Science in Banking and Finance • Data Science in Retail and E-commerce • Data Science in Education • Data Science in Government and Smart Systems
Module 2 (Credit 1) Data Preparation and Statistical Analysis	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand basic data preparation and preprocessing techniques.
	2. Apply data cleaning methods to improve data quality.
	3. Use basic statistical measures to analyze and interpret data.
Content Outline	**Data Preparation and Preprocessing** • Understanding datasets and data attributes • Data cleaning concepts • Handling missing values • Removing duplicates and inconsistent data • Data transformation and normalization (conceptual**)** **Basic Statistical Analysis** • Measures of central tendency: mean, median, mode • Measures of dispersion: range, variance, standard deviation • Introduction to correlation analysis • Interpreting statistical results for decision-making

Activities to be done in the class towards Comprehensive Continuous Evaluation
Module 1: 1. Identify and discuss real-world problems where Data Science is applied. 2. Classify datasets based on data types and sources. 3. Case study discussion on Data Science applications in different domains.
Module 2: 1. Analyze a given dataset and identify data quality issues. 2. Compute basic statistical measures manually or using tools (Excel / calculator). 3. Interpret results and present insights through short reports or presentations.

Reference:

- McKinney, W. (2017). *Python for Data Analysis*. O'Reilly Media.
- Grus, J. (2019). *Data Science from Scratch*. O'Reilly Media.
- VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly Media.
- Matthes, E. (2019). *Python Crash Course*. No Starch Press.
- Official Python Documentation – https://docs.python.org

Semester – III

.3.4 Minor Stream

Course Title	**Data Analytics**
Course Credits	**4 Credits**
Course Outcomes	1. Understand data analytics concepts and processes. 2. Analyze and preprocess datasets for meaningful insights. 3. Apply statistical and visualization techniques for analysis. 4. Support decision-making using data-driven insights.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Explain** the fundamentals of data analytics and related terminologies. 2. **Identify** different types of data and analytics methods. 3. **Describe** the data analytics lifecycle. 4. **Recognize** real-world applications of data analytics across domains.
Content Outline	➢ Introduction to Data Analytics ➢ Difference between Data Analytics, Data Science, and Business Analytics ➢ Types of Data • Structured, Semi-Structured, Unstructured ➢ Types of Analytics • Descriptive, Diagnostic, Predictive, Prescriptive ➢ Data Analytics Lifecycle ➢ Applications of Data Analytics in Real-World Domains • Business, Healthcare, Finance, Education, Social Media

Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Identify** various data sources and data collection methods. 2. **Analyze** data quality issues in real datasets. 3. **Apply** data preprocessing techniques to prepare data for analysis. 4. **Demonstrate** basic data transformation and sampling techniques.
Content Outline	➢ Data Sources • Primary and Secondary Data • Internal and External Data Sources ➢ Data Collection Methods ➢ Data Quality Issues ➢ Data Preprocessing Techniques • Data Cleaning • Handling Missing Values • Noise Reduction • Data Transformation • Data Integration ➢ Introduction to Data Sampling
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Perform** exploratory data analysis on datasets. 2. **Apply** descriptive statistical techniques to summarize data. 3. **Analyze** relationships between variables using correlation methods. 4. **Interpret** statistical results for analytical insights.

Content Outline	➢ Introduction to Exploratory Data Analysis (EDA) ➢ Descriptive Statistics • Mean, Median, Mode • Variance and Standard Deviation ➢ Data Distribution and Visualization ➢ Correlation and Covariance ➢ Introduction to Probability Concepts ➢ Basic Statistical Analysis for Data Analytics
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Create** effective data visualizations for analytical reporting. 2. **Interpret** visual and statistical insights from data. 3. **Apply** data analytics results to support decision-making. 4. **Recognize** ethical considerations in data analytics.
Content Outline	➢ Importance of Data Visualization ➢ Types of Charts and Graphs • Bar Chart, Line Chart, Pie Chart, Histogram ➢ Introduction to Visualization Tools • Excel / Google Sheets / Power BI (Conceptual) ➢ Dashboard Concepts ➢ Interpreting Analytical Results ➢ Data-Driven Decision Making ➢ Ethical Issues in Data Analytics ➢ Case Studies in Data Analytics

Reference Books:

1. **Data Analytics Made Accessible** — *Dr. Anil Maheshwari*
   Comprehensive introductory book covering key concepts, lifecycle, case studies, and examples.

2. **Python for Data Analysis** — *Wes McKinney*
   Practical guide to using Python (pandas, NumPy) for data manipulation, cleaning, and analysis. (Highly recommended for analytics tools)

3. **Practical Statistics for Data Scientists** — *Peter Bruce & Andrew Bruce*
   Covers essential statistical concepts used in analytics with examples in R/Python.

4. **Storytelling with Data: A Data Visualization Guide for Business Professionals** — *Cole Nussbaumer Knaflic*
   Focuses on effective visualization design and communicating insights clearly.

5. **The Visual Display of Quantitative Information** — *Edward R. Tufte*
   Classic on visualization principles and graphical excellence in analytics.

Semester - III

.3.7 Major (Core)

Course Title	**Cognitive Science for Artificial Intelligence**
Course Credits	**2 Credits**
Course Outcomes	1. Explain fundamental concepts of cognitive science and their relevance to Artificial Intelligence. 2. Describe human cognitive processes such as perception, attention, memory, learning, and reasoning. 3. Analyze how cognitive models and theories influence the design of intelligent systems. 4. Relate cognitive architectures to AI techniques such as problem solving, decision making, and learning.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Explain the basic concepts and scope of Cognitive Science 2. Describe key human cognitive processes 3. Understand how humans process information 4. Relate cognitive theories to Artificial Intelligence concepts
Content Outline	• Introduction to Cognitive Science • Interdisciplinary nature of Cognitive Science (Psychology, Neuroscience, AI, Linguistics, Philosophy) • Human cognition and information processing • Cognitive processes: perception, attention, memory, and learning • Cognitive theories and models • Relationship between human cognition and Artificial Intelligence

Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand different cognitive architectures and models 2. Analyze the role of cognition in AI problem solving 3. Apply cognitive principles to AI system design 4. Identify ethical and social concerns in cognitive-based AI systems
Content Outline	• Cognitive architectures (symbolic and connectionist models) • Cognitive modeling in Artificial Intelligence • Reasoning, problem solving, and decision making • Human–Computer Interaction (HCI) and user-centered design • Cognitive principles in intelligent systems • Ethical and social implications of cognitive-based AI

Course Syllabus

Semester - IV

.4.1 Major (Core)

Course Title	**Database Management System**
Course Credits	**4 Credits**
Course Outcomes	1. Apply the fundamental architecture, data models, and user roles of database management systems to real-world scenarios.
	2. Analyze entity relationships and map conceptual models into relational schemas for structured database design.
	3. Evaluate complex SQL and relational algebra queries for accuracy, performance, and business logic implementation.
	4. Apply normalization techniques and transaction management principles to maintain database consistency and reduce anomalies.
	5. Create optimized database solutions with appropriate concurrency control and indexing mechanisms based on application needs.
Module 1 (Credit 1) Introduction to Database Management system and Database Systems Architecture	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply the core concepts of data, information, and knowledge to distinguish database use-cases.
	2. Analyze the advantages of DBMS over file systems.
	3. Describe real-world DBMS applications across domains.
	4. Understand 2-tier and 3-tier architectures and abstraction levels.
	5. Identify different database models and users.

	6. Apply the core concepts of data, information, and knowledge to distinguish database use-cases.
Course Outline	➢ **Introduction to Database Systems** • **Basic Concepts**:Data, Information, and Knowledge,Database and Database Management System (DBMS),Characteristics & Advantages of DBMS,Applications of DBMS in Real-World Scenarios • **Database System Architecture**: 2-Tier Architecture & 3-Tier Architecture ,Data Abstraction Levels (Physical, Logical, View),Database Users: DBA, Developers, End Users • **Types of Database Models** : Hierarchical Model,Network Model,Relational Model (RDBMS),Object-Oriented Model
Module 2 (Credit 1) Introduction to Data Models, RDBMS and Database Design	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply ER modeling to visualize data systems.
	2. Analyze attributes, relationships, and keys to construct ER diagrams.
	3. Convert ER models into relational schemas.
	4. Evaluate integrity constraints in relational designs.
Course Outline	➢ **Entity-Relationship (ER) Model & Relational Model** • **Entity-Relationship (ER) Model** : Entities, Attributes, Relationships,Types of Attributes (Simple, Composite, Derived, Multi-Valued),Relationship Types (One-to-One, One-to-Many, Many-to-Many),Keys: Primary, Foreign, Super, Candidate,ER Diagram Notations,Mapping ER Model to Relational Model • **Relational Model Concepts** : Tables, Tuples, Attributes, AttributesRelational Schema & Constraints,Domain, Entity, Referential Integrity Constraints.

<table>
<tr><td colspan="2">Module 3 (Credit 1) Introduction to Functional dependencies, Normalization and Structured Query Language</td></tr>
<tr><td rowspan="4">Learning Outcomes</td><td>After learning this module learners will be able to understand</td></tr>
<tr><td>1. Apply relational algebra operations to retrieve and manipulate datasets.</td></tr>
<tr><td>2. Create and execute SQL queries for DDL, DML, DCL, and TCL operations.</td></tr>
<tr><td>3. Evaluate query outcomes using aggregate functions, grouping, and joins.
4. Analyze indexing and views to improve performance.</td></tr>
<tr><td>Course Outline</td><td>➢ Structured Query Language (SQL) & Relational Algebra
• Relational Algebra : Basic Operations: Selection (σ), Projection (π), Union (∪), Intersection, Difference, Cartesian Product (×),Joins (Inner, Outer, Natural, Equi-Join),Division and Assignment Operations
• SQL (Structured Query Language)
• Data Definition Language (DDL): CREATE, ALTER, DROP
• Data Manipulation Language (DML): INSERT, UPDATE, DELETE
• Data Control Language (DCL): GRANT, REVOKE
• Transaction Control Language(TCL):COMMIT, ROLLBACK, SAVEPOINT
• Aggregate Functions: COUNT, SUM, AVG, MIN, MAX GROUP BY, HAVING, ORDER BY
• Joins: INNER, LEFT, RIGHT, FULL
• Subqueries & Nested Queries Views and Indexing</td></tr>
<tr><td colspan="2">Module 4 (Credit 1) Transaction Processing System, Concurrency Control Techniques and Database Recovery System</td></tr>
<tr><td>Learning Outcomes</td><td>After learning this module learners will be able to understand</td></tr>
</table>

	1. Apply functional dependency rules to detect anomalies. 2. Analyze relations to achieve appropriate normal forms. 3. Evaluate transaction properties and implement basic control mechanisms. 4. Create database systems with concurrency solutions and deadlock strategies.
Course Outline	➢ **Normalization, Database Design, Transaction Management & Concurrency Control** • **Concept of Normalization** : Functional dependencies,Anomalies in Unnormalized Relations • **Normal Forms**: First Normal Form (1NF),Second Normal Form (2NF),Third Normal Form (3NF),Boyce-Codd Normal Form (BCNF),Fourth (4NF) & Fifth Normal Form (5NF),Lossless Join Decomposition & Dependency Preservation • **Transactions and ACID Properties** Atomicity, Consistency, Isolation, Durability (ACID),Transaction States: Active, Partially Committed, Failed, Aborted, Committed • **Concurrency Control Techniques** Problems in Concurrency: Dirty Read, Lost Update, Phantom Read,Locking Mechanisms (Shared & Exclusive Locks),Two-Phase Locking (2PL),Timestamp-Based Protocols,Deadlock Handling: Prevention, Detection, Recovery
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: **Introduction to Database Management System and Database Systems Architecture** Assignment Title: *Understanding Database Systems and Architecture* Assignment Tasks: 1. Define the following with suitable examples: • Database • DBMS • Database Applications 2. Compare and contrast: • Traditional File System vs Database System (Tabulate at least 5 differences)	

3. Describe the roles of different Database Actors (e.g., DBA, End-users, Application Programmers).

4. Explain Data Abstraction levels in DBMS with a diagram.

5. Create a diagrammatic representation of:
   - Three-Tier Architecture

6. Centralized and Client-Server Architecture

Module 2:

Introduction to Data Models, RDBMS and Database Design
Assignment Title: *Data Models and Entity Relationship Design*
Assignment Tasks:

1. Define Data Model and briefly explain:
   - Hierarchical
   - Network
   - Relational
   - Object-oriented data models

2. Write a short note on:
   - RDBMS
   - NoSQL vs Traditional Databases
   - Distributed Database

3. Explain with examples:
   - Primary Key
   - Foreign Key
   - Composite Key
   - Surrogate Key

4. Design an ER Diagram for a college database with entities: Student, Course, Faculty, Enrollment. Include:
   - Attributes
   - Keys
   - Relationships

5. Use extended ER features (if any)

Module 3:

Introduction to Functional Dependencies, Normalization and Structured Query Language

Assignment Title: *Database Normalization and SQL Implementation*
Assignment Tasks:

1. Define and explain the importance of:
   - Functional Dependencies
   - Normal Forms (1NF to BCNF)

2. Normalize the given unnormalized table to 3NF.
   (Provide sample unnormalized data: e.g., student info with repeating groups.)

3. Write SQL queries for the following tasks:
   - Create a table for "Employee" with at least 5 fields and constraints.
   - Insert 3 rows of data.
   - Write a query to display the highest salary.
   - Display employees in ascending order of joining date.
   - List departments with more than 5 employees (use GROUP BY & HAVING).

4. Perform different joins (INNER JOIN, LEFT JOIN) between Employee and Department tables.

Module 4:

Transaction Processing, Concurrency Control and Recovery System

Assignment Title: *Transactions, Concurrency, and Recovery Techniques in DBMS*
Assignment Tasks:

1. Define Transaction, ACID properties, and explain with real-time examples.
2. Differentiate between serial and concurrent schedules using diagrams.
3. Discuss the need for Concurrency Control and compare:
   a. 2PL
   b. Timestamp Ordering
   c. Optimistic Concurrency Control
4. Describe the steps of Database Recovery:
   a. Checkpoints Caching

Reference Books:

1. Korth, H. F., & Silberschatz, A. (2010). Database system concepts (6th ed.). McGraw- Hill.
2. Elmasri, R., & Navathe, S. B. (2010). Fundamentals of database systems (6th ed.). McGraw-Hill.
3. Bayross, P. (n.d.). Oracle: The complete reference. BPB Publications.

4. Datapro InfoWorld Ltd. (n.d.). Upgrade to Oracle 8.
5. Widom, J., & Wiederhold, G. (1995). Database design. McGraw-Hill.

Assessment:

Internal Assessment: (50 marks) Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Clear definitions of terms (e.g., DBMS, normalization, transaction).Demonstrates subject clarity.	10
Accuracy and Completeness	All questions are attempted. Answers are factually correct.Diagrams/SQL code are properly done.	15
Application and Analysis	Applies concepts correctly (e.g., ER diagrams, joins, normalization).Includes real-life relevance.	10
Presentation and Neatness	Organized layout. Proper use of headings, indentation, bullet points, tables, and labeled diagrams.	5
Innovation and Effort	Attempting original examples. Uses SQL screenshots, ER tools, or explains business cases practically.	5
Timely Submission	Submitted within the deadline. Late submission deducts up to 5 marks unless exempted.	5

Given below are two sample projects but it is expected to work on similar sorts of projects. (ER Diagrams, Tables and SQL Queries)

Project 1: Inventory Management System (Console-based)

Project Description:

This project is about creating a console-based Inventory Management System for a small store or warehouse. The system will help manage product details, stock levels, and sales. Users can add new products, update stock quantities, and generate reports on current

inventory and sold items. The application will store data using collections like arrays or lists, and ensure data consistency through input validation.

Project 2: Library Management System (Console-based)
Project Description:
This project involves building a console-based Library Management System to manage books, borrowers, and transactions. The system will allow the librarian (admin) to add, delete, update, and search for books. It will also manage borrowing and returning books, tracking due dates, and checking the availability status of each book.
The data will be stored in memory using collections such as arrays, lists, or dictionaries.

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester - IV

.4.2 Major (Core)

Course Title	**Advanced Python**
Course Credit	**4 Credits**
Course Outcome	1. Apply advanced Python programming concepts to develop efficient and modular applications.
	2. Implement object-oriented design principles and advanced OOP features in Python.
	3. Use advanced file handling, exception handling, and logging mechanisms effectively.
	4. Develop database-driven Python applications using SQL and Python libraries.
	5. Build concurrent, modular, and reusable Python programs for real-world problems.
Module 1(Credit 1) Advanced Python Language Features	
Learning Outcome	**After learning the modules, learners will be able to**
	1. Use advanced built-in functions and functional programming concepts
	2. Implement decorators and generators for efficient code execution
Content Outline	• Advanced Data Types and Operations • List, Dictionary, and Set Comprehensions • Functional Programming: map(), filter(), reduce() • Lambda Expressions (Advanced Use Cases) • Iterators and Generators
Module2 (Credit 1) Advanced Object-Oriented Programming	
Learning Outcome	**After learning the modules, learners will be able to**
	1. Design robust applications using advanced OOP concepts

	2. Apply design principles for reusable and maintainable code
Content Outline	• Review of OOP Concepts • Magic Methods (__init__, __str__, __repr__, etc.) • Method Overloading and Overriding • Multiple Inheritance and Method Resolution Order (MRO) • Abstract Base Classes and Interfaces • Design Patterns: Singleton, Factory (Introductory)
Module3 (Credit 1) Advanced File Handling, Exception Handling & Modules	
Learning Outcome	**After learning the modules, learners will be able to**
	1. Handle files efficiently using advanced techniques
	2. Create reusable modules and packages
Content Outline	• Advanced File Handling (Binary Files, CSV, JSON) • Context Managers (with statement) • Advanced Exception Handling and Custom Exceptions • Logging and Debugging Techniques • Python Modules and Packages • Virtual Environments and Dependency Management
Module 4(Credit 1) Concurrency, Database & Real-World Applications	
Learning Outcome	**After learning the modules, learners will be able to**
	1. Implement concurrent programs using Python
	2. Develop database-driven applications

Content Outline	• Multithreading and Multiprocessing • Synchronization and Thread Safety • Introduction to Async Programming (async / await) • Python Database Connectivity (SQLite / MySQL) • CRUD Operations using Python

Activities to be done in the classroom towards Comprehensive Continuous Evaluation (CCE)

Module 1

- Write a program to implement list, dictionary, and set comprehensions for real-world data (e.g., filtering student marks).
- Write a program using map(), filter(), and reduce() to process numerical data.
- Write a program to create and use lambda functions for sorting and filtering records.
- Write a program demonstrating the use of iterators and generators.
- Write a program to create and apply decorators for logging function execution time.

Module 2

- Write a program demonstrating the use of magic methods (__str__, __repr__, __len__).
- Write a program to demonstrate multiple inheritance and Method Resolution Order (MRO).
- Write a program using abstract base classes to implement an interface.
- Write a program to demonstrate method overloading and method overriding.
- Write a program to implement the Singleton design pattern in Python.
- Write a program to implement the Factory design pattern for object creation.

Module 3

- Write a program to read and write data using CSV files.
- Write a program to store and retrieve data using JSON files.
- Write a program demonstrating the use of context managers (with statement).
- Write a program to create and handle user-defined exceptions.
- Write a program to implement logging for error tracking and debugging.
- Write a program to create a reusable Python module and package.

Module 4

- Write a program demonstrating multithreading using the threading module.
- Write a program demonstrating multiprocessing for CPU-bound tasks.
- Write a program using async and await for asynchronous execution.

- Write a program to connect Python with a database (SQLite / MySQL).
- Write a program to perform CRUD operations using Python and SQL.

Reference Books

1. Fluent Python: Clear, Concise, and Effective Programming”"Eloquent JavaScript" – Marijn Haverbeke
2. **Python Cookbook” (3rd Edition)** — *David Beazley, Brian K. Jones*"You Don’t Know JS" series – Kyle Simpson
3. **“Effective Python: 90 Specific Ways to Write Better Python”** — *Brett Slatkin*

Assessment:

Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below). Two sample projects are give below but it is expected from students to choose a good and advanced Java project topic that aligns with their interests and abilities to work on.

Evaluation Parameters	Description / Evaluation Points	Marks
Conceptual Understanding	Clear understanding of advanced Python concepts such as decorators, generators, OOP principles, concurrency, database connectivity, and exception handling.Demonstrates strong subject clarity and terminology usage.	**10**
Accuracy and Completeness	All assigned programs / questions are attempted.Solutions are logically correct and syntactically accurate.Programs execute successfully with correct output (files, threads, database operations).	**15**
Application and Analysis	Applies advanced Python concepts to solve real-world problems.Uses appropriate programming constructs (OOP, multithreading, async, DB operations).Demonstrates analytical thinking and problem decomposition.	**10**
Presentation and Neatness	Well-organized code structure.Proper indentation, comments, meaningful variable and function names.Clear separation of modules, classes, and outputs.	**5**
Innovation and Effort	Attempts original examples or enhancements beyond basic requirements.Uses real-life case studies (student system, inventory, banking, etc.).Implements additional features such as logging, validations, or optimizations.	**5**

Evaluation Parameters	Description / Evaluation Points	Marks
Timely Submission	Assignments / lab records / mini projects are submitted within the stipulated deadline.Late submissions may attract a deduction of up to 5 marks unless officially exempted.	**5**
Total		**50 Marks**

Given below are two sample projects but it is expected to work on similar sort of projects Internal evaluation Projects

Smart Student Management System

Objective: Develop a **menu-driven Student Management System** using Advanced Python concepts. The system should manage student records efficiently with file handling and database integration.

External Assessment: (50 Marks)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester - IV

.4.3 Minor Stream

Course Title	**Machine Learning**
Course Credits	**4 Credits**
Course Outcomes	1. Understand fundamental concepts and principles of Machine Learning.
	2. Apply Python programming and libraries to implement Machine Learning algorithms.
	3. Analyze and preprocess real-world datasets using Python tools.
	4. Build, evaluate, and compare Machine Learning models.
	5. Solve real-world problems using supervised and unsupervised learning techniques.
	6. Understand challenges, limitations, and ethical considerations in Machine Learning.
Module 1 (Credit 1) Introduction to Machine Learning	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Explain the fundamentals, need, and applications of Machine Learning.
	2. Understand the general architecture and workflow of Machine Learning systems.
	3. Analyze core Machine Learning concepts such as bias, variance, and overfitting.
	4. Classify different types of Machine Learning approaches.

Content Outline	➢ **Introduction to Machine Learning** • Definition and need of Machine Learning • Knowledge-driven vs data-driven approach • Applications of Machine Learning • Advantages, disadvantages, and challenges • General architecture of Machine Learning systems ➢ **Core Concepts in Machine Learning** • Inductive learning and generalization • Bias and variance • Overfitting and underfitting • Parametric vs non-parametric algorithms ➢ **Types of Machine Learning** • Supervised learning • Unsupervised learning • Semi-supervised learning • Reinforcement learning • Introduction to Deep Learning and Neural Networks
Module 2 (Credit 1)	
Learning Outcomes	**Classification Techniques**
	After learning the module, learners will be able to
	1. Understand classification problems and their types.
	2. Implement classification algorithms using Python.
	3. Analyze decision-making techniques in classification models.
	4. Apply Artificial Neural Networks for classification tasks.
	5. Understand classification problems and their types.
Content Outline	➢ **Introduction to Classification** • Concept of classification • Binary classification • Multi-class classification • Multi-label classification

	➢ **K-Nearest Neighbour (KNN) Algorithm**
	
	• Need and working of KNN
	• Distance computation methods
	• Advantages and disadvantages
	• Implementation of KNN using Python (Scikit-learn)
	
	➢ **Decision Tree Classification**
	
	• Terminology and assumptions
	• Working of Decision Trees
	• ID3 Algorithm
	• Attribute selection methods:
	  ○ Entropy
	  ○ Information Gain
	  ○ Gini Impurity
	• Decision Tree implementation using Python
	
	➢ **Artificial Neural Networks (ANN)**
	
	• Basic structure of ANN
	• Neuron model and activation functions
	• ANN for classification (conceptual + Python-based overview)

Module 3 (Credit 1) Regression, SVM and Model Evaluation using Python	
Learning Outcomes	**After learning the module, learners will be able to** 1. Explain regression techniques and their applications. 2. Implement Support Vector Machines using Python. 3. Understand probabilistic learning approaches. 4. Evaluate Machine Learning models using standard metrics.
Content Outline	➢ **Regression Techniques** • Linear Regression • Logistic Regression • Differences between linear and logistic regression • Regression model implementation using Python ➢ **Support Vector Machines (SVM)**

	• Concept and working of SVM • Support vectors • Hard margin and soft margin • Kernel functions • Advantages and disadvantages • SVM implementation using Python ➢ **Probabilistic Learning** • Introduction to Bayesian learning • Bayes' theorem • Naïve Bayes classifier • Characteristics and applications • Naïve Bayes implementation using Python ➢ **Model Performance Evaluation** • Confusion matrix • Accuracy • Precision • Recall (Sensitivity) • F1-score • ROC and AUC curve • Cross-validation using Python
Module 4 (Credit 1) Unsupervised Learning and Dimensionality Reduction	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand the importance of unsupervised learning.
	2. Apply clustering algorithms to real-world datasets.
	3. Analyze high-dimensional data using dimensionality reduction techniques.
	4. Implement unsupervised learning techniques using Python.

Content Outline	➢ **Introduction to Unsupervised Learning** • Concept, importance, and challenges • Applications of unsupervised learning ➢ **Clustering Techniques** • Concept of clustering • Applications of clustering ➢ **Hierarchical Clustering** • Introduction to hierarchical clustering • Types of hierarchical clustering • Issues with hierarchical clustering • Implementation using Python ➢ **Partition-Based Clustering** • K-means clustering • Steps involved in K-means algorithm • Strengths and weaknesses of K-means • Issues in K-means clustering • Implementation using Python ➢ **Dimensionality Reduction** • Curse of dimensionality • Principal Component Analysis (PCA) • Importance of dimensionality reduction • PCA implementation using Python

Activities to be done in the class towards Comprehensive Continuous Evaluation
• Implementing a K-Nearest Neighbour algorithm (e.g. to classify handwritten digits) • Building a decision tree model using the ID3 algorithm (e.g. to predict whether a customer will churn or not) • Developing a Support Vector Machine model (e.g. to classify email messages as spam or not spam) • Building a Naïve Bayes classifier (e.g. to classify movie reviews as positive or negative sentiment) • Implementing linear regression (e.g. to predict housing prices based on features) • Using logistic regression (e.g. to predict whether a credit card transaction is fraud) • Evaluating a classification model using metrics such as accuracy, precision, recall, and F1 score • Applying hierarchical clustering (e.g. to group customer segments based on their purchasing behaviour) • Implementing the K-means clustering algorithm (e.g. to identify distinct clusters in a customer demographic dataset). • Utilizing Principal Component Analysis (PCA) for dimensionality reduction to improve the efficiency and interpretability of a model.

Reference Books

1. Machine Learning: Concepts, Techniques and Applications, T V Geetha S, Sendhil kumar CRC Press, Taylor and Francis, 1 st Edition, 2023

2. Machine Learning for Decision Sciences with Case Studies in Python, S. Sumathi, Suresh V. Rajappa, CRC Press, Taylor and Francis, 1 st Edition, 2022

3. Introduction to Machine Learning with Python, Andreas C. Müller, Sarah Guido O'Reilly Media, Inc. 1 st Edition, 2016

4. Machine Learning for Beginners, Harsh Bhasin, BPB, 1st Edition, 2020

5. Machine Learning S Sridhar, Oxford University, Press, 1 st Edition, 2021

6. Machine Learning Ruchi Doshi, Kamal Kant Hiran, BPB, 1 St Edition, 2021

Assessment:

Internal Assessment – 50 Marks

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Code Implementation	Accuracy in writing and executing the C program, correct logic, syntax	15
Problem Understanding	Clarity of problem interpretation and correct approach	10
Output Correctness	Program compiles without errors and gives correct output	10
Code Quality & Structure	Proper use of indentation, comments, meaningful variable names, modularity	5
Debugging Skills	Ability to identify and fix logical/syntax errors during execution	5
Lab Record/Journal	Completeness, correctness, formatting, and timely submission	5

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: Customer Churn Prediction Using Classification Algorithms
Description:
This project focuses on predicting whether a customer is likely to churn using classification algorithms such as K-Nearest Neighbour, Decision Tree, Support Vector Machine, and Naïve Bayes. The model is evaluated using performance metrics like accuracy, precision, recall, and F1-score.

Techniques Used:
KNN, Decision Tree (ID3), SVM, Naïve Bayes, Performance Evaluation Metrics

Project 2: Spam Email Detection System
Description:
Students build a machine learning model to classify emails as spam or non-spam using supervised learning algorithms. Feature extraction and model evaluation are performed to analyze the effectiveness of different classifiers.
Techniques Used:
Naïve Bayes, SVM, Classification Metrics, Cross Validation

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted by the college.

Semester - IV

.4.5 Skill Enhancement Courses (SEC)

Course Title	**AI Safety, Privacy & Security**
Course Credits	**2 Credits**
Course Outcomes	1. Understand safety, privacy, and security challenges in AI systems.
	2. Analyze ethical and regulatory issues related to AI deployment.
	3. Design AI solutions that are safe, privacy-aware, and secure.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Explain** the concept and importance of AI safety.
	2. **Identify** risks, biases, and failure modes in AI systems.
	3. **Analyze** ethical challenges related to fairness and transparency in AI.
	4. **Apply** responsible AI principles to improve safety and reliability.
Content Outline	• Introduction to AI Safety • Risks and Failures in AI Systems • Bias, Fairness, and Transparency in AI • Explainable and Interpretable AI (XAI) • Human-in-the-Loop and Responsible AI • Robustness and Reliability of AI Models • Safety in Machine Learning Systems

Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Explain** privacy concerns and data protection principles in AI systems.
	2. **Identify** common security threats and attacks on AI models.
	3. **Evaluate** privacy-preserving and security techniques for AI applications.
	4. **Apply** legal, ethical, and security considerations in AI system design.
Content Outline	➢ Introduction to Data Privacy in AI ➢ Personal Data, Sensitive Data, and Consent ➢ Privacy-Preserving Techniques • Data Anonymization and Pseudonymization • Differential Privacy (Conceptual) • Federated Learning (Overview) ➢ AI Security Threats • Adversarial Attacks • Data Poisoning • Model Inversion and Model Stealing ➢ Secure AI System Design ➢ Legal and Regulatory Frameworks • GDPR, Data Protection Laws

Semester - IV

.4.7 Minor Stream

Course Title	**Data Visualization and Data Analytics**
Course Credits	**2 Credits**
Course Outcomes	1. Analyze datasets using fundamental data analytics techniques 2. Visualize data effectively to uncover and communicate insights 3. Use common analytics and visualization tools for real-world datasets
Module 1 Introduction to Data Analytics	
Learning Outcomes	1. Understand the role and importance of data analytics in decision-making 2. Identify different types of data and data sources 3. Perform basic data cleaning and preprocessing techniques 4. Apply exploratory data analysis to uncover insights
Content Outline	➢ Introduction to Data Analytics • Definition and importance of data analytics • Types of analytics: Descriptive, Diagnostic, Predictive, Prescriptive ➢ Data Types and Data Sources • Structured, semi-structured, and unstructured data • Data sources: databases, spreadsheets, APIs, open datasets ➢ Data Collection and Preprocessing • Data cleaning, handling missing values, data transformation • Data normalization and formatting

Module 2	**Data Visualization**
Learning Outcomes	1. Create appropriate visualizations for different types of data 2. Apply visualization principles to improve clarity and readability 3. Develop basic dashboards using visualization tools 4. Interpret visual data to support analytical conclusions 5. Communicate data-driven insights effectively and ethical
Content Outline	➢ Introduction to Data Visualization • Importance of visualization in data analytics • Principles of effective visualization ➢ Types of Data Visualizations • Bar charts, line charts, pie charts, histograms, scatter plots • Choosing the right chart for the right data ➢ Visualization Tools and Techniques • Excel / Google Sheets charts • Introduction to Power BI / Tableau

Course Syllabus

Semester - V

.5.1 Major (Core)

Course Title	**Explainable AI (XAI) and Model Interpretability**
Course Credits	**4 Credits**
Course Outcomes	1. Explain the need and importance of Explainable AI.
	2. Understand different types of explainability and interpretability methods.
	3. Apply basic XAI techniques to interpret AI/ML model decisions
	4. Analyze ethical, legal, and trust-related issues in AI systems.
	5. Use explainability concepts in real-world AI applications.
Module 1 (Credit 1) Introduction to Explainable AI	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Define Explainable AI and model interpretability.
	2. Understand why explainability is important in AI systems.
	3. Identify problems with "black-box" models.
	4. Recognize real-world scenarios where explainability is required.

Content Outline	➢ What is Explainable AI (XAI)? ➢ Interpretability vs Explainability (simple comparison) ➢ Black-box models vs Transparent models ➢ Why Explainability Matters • Trust in AI • Accountability and transparency ➢ Examples of Unexplainable AI Failures ➢ Applications Requiring Explainable AI • Healthcare • Finance • Law and governance
Module 2 (Credit 1) Types of Interpretability and Explanation Methods	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Identify different levels and types of explainability.
	2. Understand global and local explanations.
	3. Explain model-specific and model-agnostic approaches.
	4. Choose suitable explanation methods for different models.
Content Outline	➢ **Types of Interpretability** • Intrinsic (built-in) interpretability • Post-hoc interpretability ➢ **Levels of Explanation** • Global explanations • Local explanations ➢ **Model-Specific vs Model-Agnostic Methods** ➢ **Simple Interpretable Models** • Linear regression • Decision trees • Rule-based systems

Module 3 (Credit 1) Popular Explainable AI Techniques	
Learning Outcomes	**After learning the module, learners will be able to** 1. Understand popular XAI techniques used in practice. 2. Interpret feature importance and model behavior. 3. Explain predictions made by ML models. 4. Apply basic XAI tools using simple examples.
Content Outline	➢ **Feature Importance** • Permutation importance ➢ **Local Explanation Techniques** • LIME (Local Interpretable Model-agnostic Explanations) • SHAP (SHapley Additive exPlanations) – basic idea ➢ **Visualization-Based Explanations** • Partial Dependence Plots • Individual Conditional Expectation (ICE) plots ➢ **Explainability in Neural Networks**
Module 4 (Credit 1) Applications, Ethics, and Future of Explainable AI	
Learning Outcomes	**After learning the module, learners will be able to** 1. Apply XAI concepts to real-world domains. 2. Understand ethical and legal requirements for explainability. 3. Analyze trust, fairness, and bias in AI models. 4. Explore future trends and careers related to XAI.
Content Outline	➢ **Explainable AI in Real-World Applications** • Healthcare diagnosis • Credit scoring and loan approval • Hiring and recruitment systems

	➢ **XAI and Ethical AI** • Fairness and bias detection • Responsible AI principles ➢ **Legal and Regulatory Aspects** • Right to explanation (introductory) • AI governance basics ➢ **Challenges and Limitations of XAI** ➢ **Future Trends in Explainable AI** ➢ **Career Roles Related to XAI**

Activities to be done in the class towards Comprehensive Continuous Evaluation

- Case study analysis of real-world AI decision systems (loan approval, medical diagnosis, hiring systems).
- Group discussion on the importance of explainability and trust in AI systems.
- Comparison activity between black-box models and interpretable models.
- Feature importance identification from sample AI model outputs.
- Role-play activity representing AI developer, user, and regulator.
- Demonstration-based exploration of Explainable AI tools (LIME / SHAP).
- Poster or infographic creation on Explainable AI concepts and applications.
- Mini project or presentation on Explainable AI use cases in different domains.
- Ethical analysis activity focusing on bias, fairness, and privacy in AI.
- Reflective writing on the need and impact of Explainable AI in society.

Reference Books:

1. **Explainable AI: Foundations, Methodologies and Applications** – Springer (covers core XAI concepts, techniques, evaluation, ethics, and applications)

2. **Explainable AI: Interpreting, Explaining and Visualizing Deep Learning** – Springer (focuses on XAI methods, visualization, and all major explainability techniques)

3. **Explainable Artificial Intelligence: A Practical Guide** by Parikshit N. Mahalle & Yashwant S. Ingle (introductory and practical overview of XAI and interpretability techniques)

4. **Explainable AI Recipes** by Pradeepta Mishra (hands-on, practical solutions for model explainability with Python)

5. **Practical Explainable AI Using Python** by Pradeepta Mishra (covers implementing XAI with Python frameworks like LIME, SHAP, etc.)

Assessment:

Internal Assessment – 50 Marks Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Code Implementation	Accuracy in writing and executing the program, correct logic and syntax	15
Problem Understanding	Clear understanding of the problem statement and appropriate approach	10
Output Correctness	Program compiles without errors and produces correct output	10
Code Quality & Structure	Proper indentation, comments, meaningful variable names, modular approach	5
Debugging Skills	Ability to identify and correct logical and syntax errors during execution	5
Lab Record/Journal	Completeness, correctness, formatting, and timely submission	5

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted by the college.

Semester - V

.5.2 Major (Core)

Course Title	**Generative AI and Applications**
Course Credits	**4 Credits**
Course Outcomes	1. Explain the fundamentals and working principles of Generative Artificial Intelligence.
	2. Differentiate between traditional AI and Generative AI models.
	3. Identify and apply Generative AI techniques for text, image, audio, and code generation.
	4. Analyze real-world applications of Generative AI across domains.
	5. Evaluate ethical, social, and legal issues related to Generative AI technologies.
Module 1 (Credit 1) Introduction to Generative Artificial Intelligence	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Define Generative AI and its objectives.
	2. Understand how Generative AI differs from traditional AI.
	3. Identify major categories of generative models.
	4. Recognize common Generative AI tools used today.

<table>
<tr><td>Content Outline</td><td>

Introduction to Artificial Intelligence
What is Generative AI?
Discriminative vs Generative Models (conceptual comparison)
Evolution of Generative AI
Types of Generative Models (overview)

Rule-based generation
Probabilistic models

Popular Generative AI Tools

ChatGPT, DALL·E, Midjourney, Copilot

Use cases and impact of Generative AI

</td></tr>
<tr><td colspan="2">Module 2 (Credit 1) Generative Models and Techniques</td></tr>
<tr><td rowspan="5">Learning Outcomes</td><td>After learning the module, learners will be able to</td></tr>
<tr><td>1. Understand the basic working of popular generative models.</td></tr>
<tr><td>2. Explain how text, image, and data generation occurs.</td></tr>
<tr><td>3. Identify strengths and limitations of different generative techniques.</td></tr>
<tr><td>4. Understand training data and model learning at a high level.</td></tr>
<tr><td>Content Outline</td><td>

Language Models and Text Generation

Tokens, prompts, and responses
Large Language Models (LLMs) – basic idea

Neural Network Basics for Generative AI
Autoencoders and Variational Autoencoders (VAE) – conceptual
Generative Adversarial Networks (GANs)

Generator and Discriminator concept

</td></tr>
</table>

	➢ **Diffusion Models (introductory overview)** ➢ **Limitations of Generative Models** • Hallucinations • Data bias
Module 3 (Credit 1) Applications of Generative AI	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Identify Generative AI applications across industries. 2. Apply prompt-based interaction for content generation. 3. Analyze benefits of Generative AI in productivity and creativity. 4. Explore domain-specific use cases.
Content Outline	➢ **Text Generation Applications** • Content writing • Chatbots and assistants ➢ **Image and Media Generation** • Image creation and editing • Video and audio synthesis (overview) ➢ **Code Generation and Software Development** • AI-assisted coding • Documentation generation ➢ **Generative AI in Education** • Personalized learning • Assessment support ➢ **Generative AI in Business and Healthcare** ➢ **Prompt Engineering Basics** • Writing effective prompts

Module 4 (Credit 1) Ethics, Challenges, and Future of Generative AI	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand ethical concerns related to Generative AI.
	2. Identify challenges and risks of misuse.
	3. Analyze social and legal implications.
	4. Explore future trends and career opportunities in Generative AI.
Content Outline	➢ **Ethical Issues in Generative AI** • Bias and fairness • Misinformation and deepfakes ➢ **Privacy, Copyright, and Data Security** ➢ **Responsible Use of Generative AI** ➢ **Limitations and Challenges** • Accuracy and reliability • Dependence on training data ➢ **Future Trends in Generative AI** ➢ **Career Opportunities and Skills for Generative AI**

Activities to be done in the class towards Comprehensive Continuous Evaluation
• Introduction activity: students list daily-life applications of Generative AI they already use. • Case study analysis of popular Generative AI tools such as ChatGPT, DALL·E, or Copilot. • Prompt-writing activity to generate text, summaries, or creative content. • Comparison activity between traditional AI systems and Generative AI models. • Group discussion on benefits and risks of Generative AI in education and workplaces. • Content generation task using Generative AI for blogs, reports, or presentations. • Image generation and interpretation activity using Generative AI tools. • Ethical analysis activity on misuse of Generative AI (deepfakes, plagiarism, misinformation). • Mini project or presentation on Generative AI applications in a chosen domain (healthcare, business, media, education).

References: -

1. **Generative AI: Techniques, Models and Applications**
   Authors: Rajan Gupta, Sanju Tiwari, and Poonam Chaudhary

2. **Introduction to Generative AI**
   Author: Numa Dhamani

3. **Generative AI in Action**
   Author: Amit Bahree

4. **Generative AI: Foundations & Concepts**
   Authors: Usman Qamar and Muhammad Summair Raza

5. **Generative AI: How ChatGPT and Other AI Tools Will Revolutionize Business**
   Author: Tom Taulli

Assessment:

Internal Assessment: (50 marks) Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

Evaluation Parameters	Description / Evaluation Points	Marks
Concept Understanding	Understanding of Generative AI concepts, models, and terminology	10
Application & Use-Case Analysis	Ability to identify and explain Generative AI applications in real-world domains	10
Prompt Design / Tool Usage	Effectiveness of prompts, appropriate use of Generative AI tools for tasks	10
Ethical Awareness	Understanding of ethical issues such as bias, misuse, copyright, and responsible use	5
Mini Project / Presentation	Relevance of topic, clarity of explanation, demonstration or case study	10
Participation & Engagement	Active participation in discussions, activities, and collaborative tasks	5
Total		**50 Marks**

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted.

Semester - V

.5.3 Indian Knowledge Systems (IKS)

Course Title	**Indian Knowledge Systems and Ethical Practices in IT**
Course Credits	**2 Credits**
Course Outcomes	1. Explain the foundational principles of Indian Knowledge Systems (IKS) such as *Dharma* (duty), *Karma* (action), and *Pramana* (means of valid knowledge), and relate these to ethical reasoning in computing and technology.
	2. Analyze ethical challenges in IT and computing (e.g., data privacy, algorithmic bias, and digital governance) using traditional Indian ethical concepts such as ***Lokasangraha*** (welfare of society) and ***Ahimsa*** (non-violence).
	3. Apply Indian moral values and ethical frameworks to formulate responsible technology practices that promote societal well-being, fairness, and accountability in IT projects.
	4. Evaluate contemporary technology issues (e.g., AI ethics, cybersecurity policies, user privacy) in light of constitutional, cultural, and moral values emphasized by Indian ethical traditions.
	5. Design guidelines and strategies for ethical decision-making in IT organizations that reflect both global standards (e.g., fairness, transparency) and Indian philosophical insights about duty and societal harmony.
Module 1 (Credit 1) Foundations of Indian Knowledge Systems (IKS) *(Philosophical and cultural roots of Indian knowledge)*	
Learning Outcomes	Students understand Indian philosophical values and their relevance to ethical thinking in IT.
Module 2 (Credit 1) Indian Contributions to Science, Mathematics & Computing *(India's legacy in scientific and logical thinking)*	
Learning Outcomes	Students appreciate India's intellectual heritage and its influence on modern computing concepts.

Course Outline	• Ancient Indian Mathematics • Zero, decimal system, place value • Contributions of Aryabhata, Brahmagupta, Bhaskaracharya • Indian Logic and Algorithms • Nyaya and Vaisheshika schools • Algorithmic thinking in ancient texts • Astronomy & Timekeeping • Indian calendars, eclipses, planetary motion • Linguistics & Computing • Panini's grammar as a precursor to formal languages • Sanskrit and structured programming logic • Traditional Engineering & Technology • Metallurgy (Iron Pillar of Delhi) • Water management systems

Suggested Pedagogy

- Case-study based learning
- Group discussions & debates
- Reflection essays
- Guest lectures from ethics / IKS experts

External Assessment: (50 Marks)

- End Semester examination of 50 marks for 2 hours duration will be conducted by college.

Semester – V

.5.4 Minor Stream

Course Title	**Advanced Analytics with AI (Predictive & Prescriptive Analytics)**
Course Credits	**4 Credits**
Course Outcomes	1. Explain advanced analytics concepts and the role of AI in data-driven decision making.
	2. Build and evaluate predictive analytics models using AI/ML techniques.
	3. Apply prescriptive analytics methods to recommend optimal decisions.
	4. Analyze real-world business and industry problems using predictive and prescriptive analytics.
	5. Evaluate ethical, operational, and deployment challenges in AI-driven analytics systems.
Module 1 (Credit 1) Foundations of Advanced Analytics and AI	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Differentiate descriptive, diagnostic, predictive, and prescriptive analytics.
	2. Understand the analytics lifecycle and data pipeline.
	3. Explain the role of AI and ML in advanced analytics.
	4. Identify suitable analytics approaches for different problem types.

<table>
<tr><td>Content Outline</td><td>

Introduction to Analytics

Descriptive vs Diagnostic vs Predictive vs Prescriptive Analytics

Analytics Lifecycle

Problem definition
Data collection, preparation, and exploration

Role of AI and Machine Learning in Analytics
Types of Data for Advanced Analytics

Structured, semi-structured, unstructured

Feature engineering basics
Tools and Platforms for Advanced Analytics
Python ecosystem, AutoML tools

</td></tr>
<tr><td colspan="2">Module 2 (Credit 1) Predictive Analytics Using AI</td></tr>
<tr><td rowspan="5">Learning Outcomes</td><td>After learning the module, learners will be able to</td></tr>
<tr><td>1. Understand predictive modeling concepts and workflows.</td></tr>
<tr><td>2. Build regression and classification models.</td></tr>
<tr><td>3. Evaluate predictive models using appropriate metrics.</td></tr>
<tr><td>4. Apply predictive analytics to real-world scenarios.</td></tr>
<tr><td>Content Outline</td><td>

Introduction to Predictive Analytics
Supervised Learning Techniques

Linear and logistic regression
Decision trees and random forests

Time Series Forecasting (introductory)

Trend, seasonality, forecasting concepts

Model Training, Validation, and Testing
Evaluation Metrics

</td></tr>
</table>

	• Accuracy, precision, recall, RMSE, MAE ➢ **Predictive Analytics Use Cases** • Sales forecasting • Customer churn prediction
Module 3 (Credit 1) Prescriptive Analytics and Optimization Techniques	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Explain prescriptive analytics and its importance. 2. Understand optimization and decision-making techniques. 3. Apply AI methods to recommend optimal actions. 4. Compare predictive and prescriptive analytics outcomes.
Content Outline	➢ **Introduction to Prescriptive Analytics** ➢ **Decision Making under Uncertainty** ➢ **Optimization Basics** • Linear programming (conceptual) • Constraint-based optimization ➢ **Simulation and Scenario Analysis** ➢ **Reinforcement Learning (high-level overview)** ➢ **Prescriptive Analytics Use Cases** • Resource allocation • Supply chain optimization • Pricing and scheduling
Module 4 (Credit 1)	**Applications, Ethics, and Advanced Topics**
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand ethical concerns related to Generative AI.
	2. Identify challenges and risks of misuse.

	3. Analyze social and legal implications. 4. Explore future trends and career opportunities in Generative AI.

Content Outline	• ☐ Advanced Analytics Applications
	  ○ Business and finance
	  ○ Healthcare analytics
	  ○ Smart cities and manufacturing
	• Decision Support Systems
	• Model Interpretability and Explainability (overview)
	• Ethical Issues in AI-Driven Analytics
	  ○ Bias, fairness, data privacy
	• Deployment Challenges
	  ○ Scalability, monitoring, model drift
	• Future Trends
	  ○ AutoML, real-time analytics

Activities to be done in the class towards Comprehensive Continuous Evaluation	
• Identify and classify real-world problems into descriptive, predictive, and prescriptive analytics. • Analyze a given dataset to define business objectives and analytics goals. • Select appropriate predictive models for different real-life use cases. • Evaluate predictive models using suitable performance metrics. • Perform what-if analysis to understand decision outcomes under changing conditions. • Formulate optimization problems by identifying objectives and constraints. • Analyze case studies of AI-based decision-making systems across industries. • Discuss ethical issues, bias, and fairness in AI-driven analytics solutions. • Develop and present a mini project applying predictive and prescriptive analytics to a real-world problem.	

References:

1. **Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking**

Authors: Foster Provost & Tom Fawcett
(Great for understanding analytics mindset, data pipelines, and predictive modeling logic)

2. **Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die**
   Authors: Eric Siegel

3. **Business Analytics: The Science of Data-Driven Decision Making**
   Author: U Dinesh Kumar

4. **Next-Gen Data Analytics and Intelligent Automation**
   Editors: Kusum Kumari Bharti & Noor Firdoos Jahan

Assessment:

Internal Assessment: (50 marks)

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

Evaluation Parameters	Description / Evaluation Criteria	Marks
Problem Understanding & Objective Definition	Clearly identifies the problem statement, analytics objective, and decision context	10
Data Analysis & Preparation	Proper understanding of dataset, data cleaning, preprocessing, and feature identification	10
Predictive / Prescriptive Model Application	Correct selection and application of predictive or prescriptive analytics techniques	10
Results & Decision Insights	Interpretation of model results and ability to derive actionable insights or recommendations	10
Presentation & Documentation	Clear explanation, structured report/presentation, visuals, and timely submission	10
Total		**50 Marks**

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted.

Semester – V

.5.5 Minor Stream

Course Title	**Deep Learning**
Course Credits	**4 Credits**
Course Outcomes	1. Understand the core principles of deep learning and artificial neural networks.
	2. Build and optimize deep learning models for various applications.
	3. Implement convolutional and recurrent architectures for structured and unstructured data.
	4. Train and fine-tune deep generative models for feature extraction and synthesis.
	5. Apply deep learning techniques to solve real-world problems in vision, NLP, and generative AI.
	6. Evaluate model performance using metrics such as accuracy, precision, recall, and loss functions.
	7. Deploy deep learning models efficiently on cloud or edge devices.
	8. Understand the ethical implications of AI and deep learning applications.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Explain the evolution, significance, and key applications of deep learning** across various industries.
	2. **Apply fundamental mathematical concepts** from linear algebra, including vectors, matrices, tensors, eigenvalues, norms, and matrix operations, in deep learning models..
	3. **Understand numerical computation issues**, such as overflow, underflow, conditioning, and algorithmic stability.
	4. **Describe the structure and working of neural networks**, including neurons, activation functions, and loss functions.
	5. **Explain the training process of neural networks**, including forward and backward propagation and gradient descent methods.

	6. **Analyze overfitting and underfitting** and apply regularization techniques such as L1, L2, and dropout to improve model performance.
Content Outline	• **Foundations of Deep Learning Introduction:** Evolution and significance of deep learning, Key applications across industries • **Mathematical Foundations: Linear Algebra:** Scalars, Vectors, Matrices, and Tensors, Matrix operations: Multiplication, Identity, and Inverses, Eigenvalues and Eigenvectors, Norms and special matrices • **Numerical Computation:** Issues of overflow and underflow, Conditioning and stability in algorithms • **Neural Networks Basics:** Structure and function of neurons, Activation functions: Sigmoid, Tanh, ReLU, and variants, Loss functions: MSE, Cross-Entropy • **Training Neural Networks:** Forward and backward propagation, Gradient Descent and its variants, Overfitting and underfitting, Regularization techniques: L1, L2, Dropout
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Explain deep feedforward network architectures,** training challenges, and weight initialization strategies.
	2. **Apply regularization techniques** such as batch normalization and dropout to improve deep network training and generalization.
	3. **Understand optimization techniques for deep networks,** including advanced optimizers (RMSprop, Adam), learning rate scheduling, and methods to handle vanishing and exploding gradients.
	4. **Describe the working principles of Convolutional Neural Networks (CNNs),** including convolution, feature maps, pooling, and popular architectures like LeNet, AlexNet, VGG, and ResNet.
	5. **Explain sequence modeling using Recurrent Neural Networks (RNNs)** and advanced variants such as LSTM,

	GRU, and bidirectional RNNs.
	6. **Understand attention mechanisms** and their role in improving performance for sequence-based deep learning applications.
Content Outline	• **Advanced Neural Network Architectures** • **Deep Feedforward Networks:** Architecture and training challenges, Weight initialization strategies • **Regularization Techniques:** Batch Normalization, Dropout and its impact on training • **Optimization for Deep Networks**: Advanced optimizers: RMSprop, Adam, Learning rate scheduling, Handling vanishing and exploding gradients • **Convolutional Neural Networks (CNNs):** Convolution operations and feature maps Pooling mechanisms, Popular CNN architectures: LeNet, AlexNet, VGG, ResNet • **Recurrent Neural Networks (RNNs):** Sequence modeling and applications, Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), Bidirectional RNNs and attention mechanisms
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Explain autoencoders and representation learning techniques,** including undercomplete, sparse, denoising, and variational autoencoders.
	2. **Understand generative modeling concepts**, including Generative Adversarial Networks (GANs), conditional GANs, and their practical applications.
	3. **Describe advanced sequence modeling techniques,** including advanced RNN architectures, transformer models, and attention mechanisms.

	4. **Explain the fundamentals of deep reinforcement learning,** including Deep Q-Networks and policy gradient methods.
	5. **Apply deep learning models to real-world applications** in natural language processing, computer vision, and speech processing.
	6. Analyze the suitability of specialized deep learning models for different problem domains and applications.
Content Outline	• **Specialized Deep Learning Models and Applications** • **Autoencoders and Representation Learning:** Undercomplete and sparse autoencoders, Denoising and variational autoencoders • **Generative Models:** Generative Adversarial Networks (GANs), Conditional GANs and applications • **Sequence Modeling**: Advanced RNN architectures, Transformer models and attention mechanisms • **Deep Reinforcement Learning:** Fundamentals of reinforcement learning, Deep Q-Networks and policy gradients • **Applications of Deep Learning:** Natural Language Processing: Text generation, translation, Computer Vision: Image classification, object detection, Speech Recognition and synthesis
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. **Explain ethical issues in deep learning**, including bias, fairness, interpretability, and explainability of AI models.
	2. **Analyze the societal and technical impact of ethical challenges** in the design and deployment of deep learning systems.
	3. **Understand scalability and deployment considerations,** including model optimization and hardware requirements such as GPUs and TPUs.

	4. **Evaluate real-world deep learning case studies** to identify best practices and lessons learned from industry applications.
	5. **Discuss emerging trends and future directions in deep learning**, including new architectures, quantum computing, and neuromorphic computing
Content Outline	• **Ethics and Practical Considerations in Deep Learning** • **Ethical Considerations:** Bias and fairness in AI models, Interpretability and explainability • **Scalability and Deployment:** Model optimization for deployment, Hardware considerations: GPUs, TPUs • **Case Studies:** Analysis of real-world deep learning applications, Lessons learned from industry implementations • **Future Directions:** Emerging Architectures, Quantum computing and deep learning, Neuromorphic computing

Activities to be done in the class towards Comprehensive Continuous Evaluation

1. Build and train a simple neural network for classification tasks.
2. Apply dropout and batch normalization to improve model performance.
3. Develop a CNN for image classification using a dataset like CIFAR-10.
4. Implement an RNN for sequence prediction tasks.
5. Construct an autoencoder for image denoising applications.
6. Develop a GAN to generate synthetic images.
7. Implement a transformer-based model for language translation.
8. Experiment with different hyperparameters to optimize model performance.
9. Deploy a trained model using a web framework for real-time inference.

Reference Books

1. Goodfellow, Ian, et al. Deep Learning. MIT Press, 2016.

2. Chollet, François. Deep Learning with Python. Manning Publications, 2018.

3. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

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

5. Aggarwal, Charu C. Neural Networks and Deep Learning: A Textbook. Springer, 2018

Internal Assessment – 50

Marks Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Problem Understanding	Clarity in understanding the problem statement, correct identification of objectives, dataset selection, and suitability of deep learning approach	10
Model Implementation	Correct implementation of deep learning model (CNN / RNN / Autoencoder / GAN / Transformer), appropriate architecture, and correct use of libraries	15
Training & Output Correctness	Proper training of the model, correct execution without errors, valid outputs, and meaningful results	10
Model Evaluation & Analysis	Use of appropriate evaluation metrics (accuracy, loss, precision, recall, etc.), interpretation of results, and comparison of outcomes	5
Code Quality & Documentation	Clean, well-structured code with proper indentation, comments, modularity, and reproducibility	5
Project Report / Lab Record	Completeness of report, clarity of explanation, formatting, screenshots/outputs, and timely submission	5

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: Image Classification Using Convolutional Neural Networks

Description:
This project focuses on building and training a Convolutional Neural Network (CNN) for image classification tasks using standard datasets such as CIFAR-10 or MNIST. Students analyse the impact of different CNN architectures and regularization techniques on model performance.

Techniques Used:
CNN, Convolution and Pooling Layers, Dropout, Batch Normalization, Performance Metrics

Project 2: Sentiment Analysis Using Recurrent Neural Networks

Description:
Students develop a deep learning model to perform sentiment analysis on text data such as movie reviews or social media posts. The project uses Recurrent Neural Networks (RNNs) and LSTM/GRU architectures to capture sequential patterns in text.

Techniques Used:
RNN, LSTM / GRU, Text Preprocessing, Accuracy and Loss Evaluation

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted by the college

Semester – V

.5.6 Vocational Skill Courses (VSC-S4)

Course Title	**Big Data Analytics**
Course Credit	**2 Credits**
Course Outcomes	**After successful completion of the course, learners will be able to:** 1. To provide an overview of the big data platforms, its use cases and Hadoop ecosystem. 2. To introduce programming skills to build simple solutions using big data technologies such as MapReduce, Scripting for No SQL and R 3. To learn the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability. 4. To enable students to have skills that will help them to solve complex real-world problems for decision support**.**
Module 1 (Credit 1) Introduction to Big Data and Hadoop	
Learning Outcomes	**After learning this module, learners will be able to:** 1. Understand the building blocks of Big Data Analytics. 2. Understand the building blocks of Big Data Analytics. 3. Explain the characteristics and types of Big Data. 4. Differentiate between traditional data processing and Big Data approaches. 5. Describe Hadoop architecture and its ecosystem components.
Content Outline	• Introduction to Big Data • Big Data Characteristics and Types of Big Data • Traditional vs Big Data Business Approach • Case Study of Big Data Solutions

	• Concept of Hadoop • Core Hadoop Components • Hadoop Ecosystem
Module 2 (Credit 1) Hadoop HDFS and MapReduce	
Learning Outcomes	**After learning this module, learners will be able to:** 1. Understand distributed file systems and HDFS architecture. 2. Explain MapReduce programming model and execution flow. 3. Apply MapReduce techniques to solve data processing problems. 4. Analyze limitations of Hadoop and MapReduce**.**
Content Outline	• Distributed File Systems • Physical Organization of Compute Nodes • Large Scale File System Organization • MapReduce Model: Map Tasks, Reduce Tasks, Combiners • MapReduce Execution and Fault Tolerance • Algorithms using MapReduce – Matrix Vector Multiplication – Relational Algebra Operations – Selection, Projection, Union, Intersection, Difference • Hadoop Limitations

Reference Books

1. Cre Anand Rajaraman and Jeff Ullman —Mining of Massive Datasets, Cambridge University Press
2. Alex Holmes —Hadoop in Practice, Manning Press, Dreamtech Press.
3. Dan Mcary and Ann Kelly —Making Sense of NoSQL – A guide for managers and the rest of us, Manning Press.
4. DT Editorial Services, —Big Data Black Book, Dreamtech Press
5. EMC Education Services, Data Science and Big Data Analytic, Wiley

Assessment:

Internal Assessment: (Marks 50)

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks.

The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	**Understanding of core subject Concepts**	**10**
Problem Analysis	**Appropriate problem identification and justification**	**10**
Technical Design	**Logical design / workflow (conceptual)**	**10**
Innovation & Originality	**Creativity in solution approach**	**5**
Communication & Presentation	**Clarity and organization**	**5**
Team Collaboration	**Active participation and coordination**	**5**
Timeliness & Completeness	**On-time and complete submission**	**5**

Semester – V

.5.7 Major (Core)

Course Title	**Human Centered AI**
Course Credit	**2 Credits**
Course Outcomes	**After successful completion of the course, learners will be able to:** 1. Understand the principles and importance of human-centered approaches in AI system design. 2. Analyze ethical, social, and usability issues in AI systems. 3. Apply human-centered design concepts to evaluate AI applications. 4. Design basic AI solutions that prioritize user needs, fairness, transparency, and trust.
Module 1 (Credit 1) Foundations of Human-Centered AI	
Learning Outcomes	**After learning this module, learners will be able to:** 1. Explain the concept and goals of Human-Centered AI. 2. Identify human needs and values in AI system development. 3. Recognize ethical and social challenges in AI adoption.
Content Outline	• Introduction to Human-Centered AI • Difference between human-centered and technology-centered AI • Human values in AI: usability, accessibility, inclusivity • Human-AI interaction fundamentals • Ethics in AI: fairness, accountability, transparency • Social impact of AI systems
Module 2 (Credit 1) Designing and Evaluating Human-Centered AI Systems	

Learning Outcomes	**After learning this module, learners will be able to:** 1. Apply human-centered design principles to AI systems. 2. Evaluate AI systems based on user experience and trust. 3. Identify challenges and future directions of Human-Centered AI.
Content Outline	• Human-Centered Design (HCD) process • User research and requirement gathering for AI systems • Explainability and trust in AI • Bias detection and mitigation from a human perspective • Evaluation methods for human-centered AI (usability testing, feedback loops) • Responsible and inclusive AI design practices • Future trends and challenges in Human-Centered AI

References:

- **Human+Machine: Reimagining Work in the Age of AI**
  Authors: Paul R. Daugherty & H. James Wilson
- **Designing Human-Centered AI**
  Authors: Andy Chu, Kriti Sharma, and Samidh Chakrabarti
- **The AI Economy: Work, Wealth and Welfare in the Robot Age**
  Author: Roger Bootle

Assessment:

Internal Assessment: (Marks 50) Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

Evaluation Parameters	Description / Evaluation Criteria	Marks
Understanding of Human-Centered AI Concepts	Demonstrates clear understanding of human-centered principles, user needs, ethics, and social impact of AI	10
Problem Identification & Human Context	Correctly identifies the problem and incorporates human values such as usability, fairness, inclusivity, and trust	10
Application of HCAI Principles	Applies human-centered design, ethical reasoning, and explainability concepts to the given AI scenario	10
Analysis & Critical Thinking	Evaluates AI systems for bias, transparency, accountability, and user impact with logical reasoning	10
Presentation / Documentation	Clarity of explanation, structured report or presentation, examples, and timely submission	10
Total		**50 Marks**

Course Syllabus

Semester - VI

.6.2 Community Engagement (CE)

Course Title	**Digital Literacy and Community Outreach**
Course Credits	2 Credits
Course Outcomes	1. Understand basic digital literacy concepts and the importance of digital skills in daily life. 2. Use digital tools safely and responsibly for communication, information access, and services. 3. Create awareness about digital plat forms and e-governance services among the community. 4. Apply digital knowledge to support community outreach and social inclusion initiatives.
Module 1 (Credit 1) Digital Literacy Fundamentals	
Learning Outcomes	1. Understand fundamental digital literacy concepts and digital devices. 2. Use basic digital tools for communication and information access. 3. Practice safe, ethical, and responsible use of digital technologies.
Course Outline	➢ **Introduction to Digital Literacy** • Meaning and importance of digital literacy • Digital divide and digital inclusion ➢ **Digital Devices and Platforms** • Computers, smartphones, tablets • Operating systems and basic applications (conceptual) ➢ **Internet Basics** • Internet services and applications • Email, search engines, and social media (overview)

	➢ **Digital Safety and Ethics** • Cyber safety and security • Password management and online privacy • Cyberbullying and responsible digital behaviour ➢ **Digital Payments and Services (Overview)** • UPI, online banking, and digital wallets • Introduction to e-governance services
Module 2 (Credit 1) Community Outreach and Digital Empowerment	
Learning Outcomes	1. Apply digital literacy skills to community-based activities. 2. Promote awareness of digital services and government initiatives. 3. Plan and participate in community outreach programs using digital tools.
Course Outline	➢ **Digital Empowerment and Society** • Role of digital literacy in social development • Digital inclusion for rural and underserved communities ➢ **E-Governance and Public Digital Services** • Digital India initiatives (overview) • Online public services: education, healthcare, banking, and welfare schemes ➢ **Community Outreach Programs** • Planning digital awareness campaigns • Conducting workshops and training sessions • Use of digital media for community engagement ➢ **Social Responsibility and Ethics** • Ethical use of digital platforms • Accessibility and inclusivity in digital systems ➢ **Case Studies** • Successful digital literacy and community outreach initiatives

Module 1

- Identify common digital tools used in daily life and demonstrate their usage.
- Create a short presentation on cyber safety and responsible internet use.
- Conduct a quiz or worksheet on digital literacy concepts.

Module 2

- Plan a digital awareness program for a local community (conceptual).
- Prepare posters or digital content promoting digital payments or e-governance services.
- Write a brief report on the impact of digital literacy in community development

Assessment

Internal Assessment: (50 Marks)

Evaluation Scheme:

The Internal Assessment shall be conducted based on **Project Development, Presentation, and Report Writing**, evaluated using a well-defined rubric.
Students are required to **select and develop one appropriate project** related to the course, demonstrating practical understanding and application of concepts learned.
Two sample project topics are provided for reference; however, students are encouraged to choose **a relevant and sufficiently advanced project topic** aligned with their interests and capabilities, subject to faculty approval.

The project evaluation will be carried out through **project demonstration and viva/presentation**, based on the following rubric:

Evaluation Criteria	Description	Marks
1. Core Functional Implementation	Correct implementation of core features, completeness of functionality, and logical flow of the application	**20**
2. User Interface Design & Responsiveness	Quality of user interface design, usability, layout consistency, and responsiveness (where applicable)	**10**
3. Code Quality & Structure	Code organization, readability, use of meaningful names, modularity, and adherence to coding standards	**10**
4. Input Validation & Error Handling	Proper handling of user input, validation techniques, and error/exception management	**5**

5. Project Documentation / Report	Quality of project report/documentation including problem statement, methodology, screenshots, results, and conclusions	**5**

Semester 1-Instead of Logic and reasoning Swap C and C++
Semester-2 Python major Introduction to AI minor
Semester 3- Maths subject instead COA
Semester 4- Advanced Python
semester 6-