AIDA AI BSc/MSc Curriculum

Ioannis Mademlis, Ioannis Pitas
Department of Informatics, Aristotle University of Thessaloniki, Greece

Date: June 2025

Contributors: I. Mademlis (Aristotle University of Thessaloniki), C. Papaioannidis (Aristotle University of Thessaloniki), X. Ziouvelou (National Center of Scientific Research "Demokritos"), R. Tolosana Calasanz (University of Zaragoza), A. Hrckova (Kempelen Institute of Intelligent Technologies), J. Renoux (Orebro University), B. O'Sullivan (University College Cork), N. Nikolaidis (Aristotle University of Thessaloniki), C. Ouzounis (Aristotle University of Thessaloniki), E. Valveny Llobet (Universitat Autònoma de Barcelona), N. Cesa-Bianchi (Università degli Studi di Milano and Politecnico di Milano), E. Gavves (University of Amsterdam), J. Peters (TU Darmstadt), P. Drineas (Purdue U), I. Pitas (Aristotle University of Thessaloniki).

Abstract

This undergraduate Artificial Intelligence curriculum template for dedicated AI academic programs/departments provides a coherent, modular pathway that conforms to contemporary higher education standards. It is organized similarly to the ACM/IEEE-CS/AAAI "CS2023" curriculum on AI. Five Knowledge Areas (Pre-Foundational CS, Advanced Mathematics & Cognitive Foundations, Core AI, Advanced & Specialized AI Electives, and AI Support & Professional Practice) map the material onto a flexible 120-credit U.S. bachelor's or 240-ECTS Bologna degree, leaving ample space for additional general education courses, internships, etc. The template is future-proofed through explicit coverage of 2025 trends in AI, which are now taught at Ivy League programs. Trustworthiness, ethics and governance are woven throughout, equipping graduates to design compliant, human-centered solutions from day one. The modular hour bands support multiple delivery models, so that institutions of varying size can adopt, adapt and scale offerings without compromising academic depth or accreditation alignment.

This Artificial Intelligence curriculum can also serve as a basis for offering postgraduate AI programs at MSc, PhD or even at postdoc level as well as for conversion AI programs for personnel reskilling/upskilling.

Introduction

Artificial Intelligence has moved from a niche research field to a technology driving national talent initiatives and shaping global regulation. As universities are under pressure to graduate practitioners who can design, deploy and govern AI at scale, the present **undergraduate AI curriculum template** targets dedicated AI academic programs. It satisfies the quantitative and qualitative thresholds set by ABET's 2025 criteria for computing programs, but it specially designed for dedicated AI departments. It further reflects the ethical imperatives codified by UNESCO and operationalized by the EU AI Act, ensuring that every graduate understands risk, accountability and human-centric design alongside algorithms, tools and data. The framework lets departments choose depth tracks that mirror those at leading universities, while permitting smaller universities to meet the same competencies through carefully selected subsets and shared online laboratories.

The AI Curriculum is organized in a manner similar to the ACM/IEEE-CS/AAAI "CS2023" curriculum on AI. It is structured around Knowledge Areas (KAs), Knowledge Units (KUs), recommended contact hours, and illustrative learning outcomes – while fully integrating:

1. The preliminary topics needed for a complete AI undergraduate program.
2. A reasonable approach to "hours per course", so that each KU can map onto one or more semester-length courses (≈45-60 contact hours each), or be combined as needed.

The U.S. system of 120 credits for 4-year Bachelor programs is assumed below. The KUs here can be combined or distributed into multiple semester courses. The "recommended hours" are approximate minimum in-class contact hours for that KU. Departments can structure them into 3-credit/4-credit courses, half-semester modules, or specialized tracks.

The curriculum is divided into **five Knowledge Areas (KAs)**:

1. **KA1: Preliminary & Foundational CS (PFCS)** - Programming paradigms, discrete math, data structures, etc.
2. **KA2: Advanced Mathematics & Cognitive Foundations (AMCF)** - Calculus, probability, optimization, plus cognitive/neuroscience fundamentals.
3. **KA3: Core AI (CAI)** - Foundations of AI, Knowledge Representation, Reasoning & Planning, Machine Learning, Deep Learning, Computer Vision, NLP, AI Ethics & Governance.
4. **KA4: Advanced & Specialized AI Electives (ASAI)** - Reinforcement Learning, Generative AI, Trustworthy AI, Robotics, etc.
5. **KA5: AI Support & Professional Practice (AISP)** - Data management, parallel/distributed programming, project management, entrepreneurship.

Each KA contains **Knowledge Units (KUs)**, each with:

- **Recommended Contact Hours** (total minimal classroom contact, including lectures and labs).
- **Topics**.
- **Key Learning Outcomes**.

This approach allows institutions flexibility in how they group the material into semester-length courses. The first three KAs (PFCS, AMCF, and CAI) contain **mandatory** topics, while KA4 (Advanced & Specialized AI Electives) and KA5 (AI Support & Professional Practice) contain **elective** topics.

KA1: Preliminary & Foundational CS (PFCS)

Description: Core computer science background needed before specialized AI courses, including structured programming, object-oriented programming, functional programming, theory of computation, discrete mathematics, data structures, computer systems and algorithms. Contains 6 KUs.

Recommended Total Hours for KA1: ~**215-300 contact hours** (e.g., courses such as "Introduction to Programming", "Discrete Mathematics", "Data Structures & Algorithms", "Theory of Computation", "OOP & Functional Programming", etc.)

PFCS-1: Intro to Programming & Structured Programming

- **Hours**: 30-45
- **Topics**:
  - Basic data types, variables, and control structures
  - Functions, modular design, parameter passing
  - Basic debugging, testing, and error handling
  - I/O operations
  - Version control essentials (e.g., Git)
- **Learning Outcomes**:
  - **Content/Knowledge**
    - (*Remember*) **Identify** fundamental constructs (variables, loops, conditionals) in a structured program.
    - (*Understand*) **Explain** how modular design principles improve readability and maintainability.
  - **Methodological/Skills**
    - (*Apply*) **Implement** small-scale programs (e.g., menu-driven console apps) that use structured paradigms.
    - (*Analyze*) **Debug** and **test** programs systematically using logs or an IDE debugger.
  - **Transferrable/Application**
    - (*Apply*) **Use** a version control system to collaborate on a simple programming project.

- (*Evaluate*) **Assess** code clarity and style, documenting best practices (naming conventions, comments).

PFCS-2: Object-Oriented Programming

- **Hours**: 30-45
- **Topics**:
  - Classes, objects, encapsulation, inheritance, polymorphism
  - Design patterns (e.g., factory, singleton, observer)
  - UML basics, code refactoring strategies
  - Exception handling in OOP languages
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Describe** how OOP concepts (inheritance, polymorphism) differ from procedural approaches.
    - *(Remember)* **List** common design patterns and outline typical scenarios for each.
  - **Methodological/Skills**
    - *(Apply)* **Develop** a moderate-scale OOP application (e.g., a school management system) using classes/interfaces.
    - *(Analyze)* **Compare** inheritance VS composition in a given design, discussing trade-offs for maintenance.
  - **Transferrable/Application**
    - *(Evaluate)* **Review** and refactor an existing OOP codebase to improve modularity and extensibility.
    - *(Create)* **Collaborate** in a team to implement a small OOP project with version control and code reviews.

PFCS-3: Discrete Mathematics & Theory of Computation

- **Hours**: 45-60
- **Topics**:
  - Mathematical logic (propositional, predicate), proof techniques (induction, contradiction)
  - Sets, relations, combinatorics, counting principles
  - Chomsky hierarchy, Turing Machines, decidability, halting problem
  - Complexity classes (P, NP, NP-complete), reductions
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** the role of combinatorics and discrete structures in algorithm analysis.
    - *(Remember)* **Identify** examples of classic NP-complete problems (SAT, traveling salesman).
  - **Methodological/Skills**

    - *(Apply)* **Prove** simple propositions using induction (e.g., correctness of a recursive algorithm).
    - *(Analyze)* **Classify** decision problems by complexity class, using polynomial-time reductions when relevant.
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** whether a real-world problem is tractable or likely requires approximation/heuristics.
    - *(Create)* **Construct** formal proofs or arguments about algorithm correctness using discrete math tools.

PFCS-4: Data Structures & Analysis of Algorithms

- **Hours**: 45-60
- **Topics**:
  - Arrays, lists, stacks, queues, trees (BST, AVL), heaps, graphs
  - Algorithmic complexity (Big-O, Big-Θ, Big-Ω)
  - Sorting (Quicksort, Mergesort, Heapsort), searching (binary search)
  - Greedy algorithms, dynamic programming, recursion VS iteration
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Remember)* **Identify** time and space complexity for fundamental data structures/operations.
    - *(Understand)* **Explain** how dynamic programming improves efficiency over naive recursion in certain tasks.
  - **Methodological/Skills**
    - *(Apply)* **Implement** at least two different sorting algorithms and analyze their runtime on sample data.
    - *(Evaluate)* **Compare** data structures (array VS linked list, BST VS heap) for a given application scenario.
  - **Transferrable/Application**
    - *(Analyze)* **Select** an optimal data structure for a real-world use case (e.g., priority queue for scheduling).
    - *(Create)* **Design** a custom data structure combining multiple concepts (hash map + tree) to solve a unique problem.

PFCS-5: Computer Systems & Architecture

- **Hours**: 45-60
- **Topics**:
  - Digital logic basics, CPU organization, memory hierarchy (cache, RAM, disks), SIMD
  - Operating systems: kernel types, processes, IPC, threads, synchronization, concurrency, filesystems
  - CPU - GPU - TPU: basic architectural differences
  - Networking fundamentals (OSI layers, protocols, topologies, TCP/IP)
- **Learning Outcomes**:
  - **Content/Knowledge**

- *(Understand)* **Explain** how CPU pipelines and caches impact program performance.
- *(Remember)* **Outline** the differences between processes and threads, discussing concurrency issues.

- **Methodological/Skills**
  - *(Apply)* **Implement** a small concurrent program with synchronization primitives (mutexes, semaphores).
  - *(Analyze)* **Compare** the performance of a parallelizable task on CPU VS GPU or TPU.
- **Transferrable/Application**
  - *(Evaluate)* **Recommend** hardware resources or configurations for AI workloads under budget constraints.
  - *(Create)* **Optimize** a system-level parameter (e.g., scheduling policy) for improved AI application performance.

PFCS-6: Functional Programming

- **Hours**: 20-30 (can be half-semester or integrated)
- **Topics**:
  - Higher-order functions, lambda calculus concepts
  - Immutability, recursion, lazy evaluation
  - Comparing functional VS imperative paradigms
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Describe** advantages of pure functions in reducing side effects for concurrent/parallel tasks.
    - *(Remember)* **Recall** how lazy evaluation differs from eager evaluation.
  - **Methodological/Skills**
    - *(Apply)* **Write** functional programs in a language like Haskell or Scala, using higher-order functions.
    - *(Analyze)* **Compare** solutions to the same problem implemented in an OOP VS functional style.
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** the maintainability and correctness benefits of immutability in large codebases.
    - *(Create)* **Integrate** small functional modules into a multi-paradigm AI pipeline (e.g., data transformations).

KA2: Advanced Mathematics & Cognitive Foundations (AMCF)

Description: Advanced math topics required for AI (calculus, differential equations, linear algebra, probability, statistics, optimization, graph/network theory, signal processing), plus cognitive psychology and neuroscience fundamentals. Contains 7 KUs.

Recommended Total Hours for KA2: ~**285-390 contact hours** (e.g., courses such as "Calculus & Differential Equations", "Probability & Statistics", "Convex Optimization", "Signal Processing", "Cognitive Psychology", "Neuroscience for AI", etc.)

AMCF-1: Calculus & Differential Equations

- **Hours**: 45-60
- **Topics**:
  - Single/multi-variable calculus (limits, continuity, derivatives, integrals)
  - Partial derivatives, gradient and Hessian
  - Ordinary Differential Equations (ODEs), basic PDEs
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** how gradients and Hessians are used in optimization-based machine learning.
    - *(Remember)* **List** standard methods for solving first-order ODEs (separation of variables, integrating factor).
  - **Methodological/Skills**
    - *(Apply)* **Solve** basic ODEs that appear in continuous-time dynamical systems for control or RL.
    - *(Analyze)* **Demonstrate** partial derivatives on multi-variable cost functions (e.g., MSE in linear regression).
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** whether a phenomenon can be approximated by linear ODEs VS requiring advanced PDE models.
    - *(Create)* **Model** a real-world system (e.g., population growth, chemical reactions) using differential equations and interpret the results for AI applications.

AMCF-2: Linear Algebra & Graph/Network Theory

- **Hours**: 45-60
- **Topics**:
  - Vector spaces, matrices, transformations, eigenvalues/eigenvectors, SVD
  - Graph theory basics (paths, connectivity, bipartite, MST, flows)
  - Algebraic graph theory basics
  - Network metrics (centrality, modularity), community detection
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Describe** how matrix factorizations (e.g., SVD, PCA) are used for dimensionality reduction.

        - *(Remember)* **Recall** common graph algorithms (BFS, DFS, Dijkstra) and their typical complexities.
    - **Methodological/Skills**
        - *(Apply)* **Compute** eigenvalues/eigenvectors for principal component analysis on a dataset.
        - *(Analyze)* **Design** a flow algorithm or community detection approach for a social network problem.
    - **Transferrable/Application**
        - *(Evaluate)* **Compare** multiple graph-theoretic approaches (PageRank, centrality measures) for ranking in a Web search.
        - *(Create)* **Develop** a method based on matrix factorization or graph embedding for a domain-specific AI challenge (e.g., recommendation systems).

AMCF-3: Probability Theory & Multivariate Statistics

- **Hours**: 45-60
- **Topics**:
    - Random variables, probability distributions (Bernoulli, Binomial, Gaussian, etc.), Law of Large Numbers, Central Limit Theorem
    - Bayes' theorem, Bayesian statistics, conditional probability, independence
    - Estimation methods, hypothesis testing, confidence intervals
    - Multivariate distributions, covariance matrices, correlation
    - Stochastic processes, measure theory basics
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** the central limit theorem and its significance for large-sample ML analysis.
        - *(Remember)* **Identify** differences among frequentist, Bayesian, and likelihood-based inference methods.
    - **Methodological/Skills**
        - *(Apply)* **Perform** hypothesis testing and confidence-interval construction on real datasets (e.g., AB testing).
        - *(Analyze)* **Model** a dataset with a multivariate Gaussian distribution, checking covariance for feature correlation.
    - **Transferrable/Application**
        - *(Evaluate)* **Interpret** correlation VS causation in an AI-driven analysis (e.g., confounding factors).
        - *(Create)* **Design** sampling or resampling (e.g., bootstrap) strategies for robust model evaluation.

AMCF-4: Information & Coding Theory, Numerical Analysis

- **Hours**: 45-60
- **Topics**:

  - Shannon entropy, channel capacity, Huffman coding, Mutual Information, KL divergence
  - Numerical methods for root finding (Newton-Raphson), polynomial interpolation
  - Least squares optimization, matrix decompositions, numerical methods for solving linear systems (Gauss-Seidel)
  - Floating-point arithmetic, stability, error analysis
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Describe** how entropy relates to data compression and the minimum bits needed for representation.
    - *(Remember)* **List** key numerical methods (Newton's method, bisection) and compare convergence rates.
  - **Methodological/Skills**
    - *(Apply)* **Implement** iterative solvers for linear or nonlinear systems and evaluate numerical stability.
    - *(Analyze)* **Examine** floating-point errors and condition numbers in large-scale matrix computations.
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** how data compression (entropy encoding) might affect subsequent ML tasks.
    - *(Create)* **Design** a custom variation of a lossless compression method and analyze its efficiency on image/audio data.

AMCF-5: Convex & Non-Convex Optimization

- **Hours**: 45-60
- **Topics**:
  - Convex sets, gradient descent, second-order methods (Newton, quasi-Newton)
  - Regularization (L1, L2), duality, constraints, Lagrange multipliers, KKT conditions
  - Non-convex optimization: SGD variants, momentum, Adam, local minima/saddle points
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** why convex optimization guarantees global optima, while non-convex does not.
    - *(Remember)* **List** common optimizers for neural networks (SGD, RMSProp, Adam) and key hyperparameters.
  - **Methodological/Skills**
    - *(Apply)* **Formulate** a linear or logistic regression problem as a convex optimization task.
    - *(Analyze)* **Interpret** gradient descent convergence behaviors VS Adam or momentum-based optimizers in deep learning.
  - **Transferrable/Application**

- *(Evaluate)* **Compare** training speed and final performance for different optimizers in a real ML scenario.
- *(Create)* **Develop** an interactive visualization where users can select different functions and see how optimization algorithms (e.g., gradient descent) navigate their landscapes.

AMCF-6: Signals & Systems

- **Hours**: 30-45
- **Topics**:
  - Signals, LTI systems, convolution, impulse response
  - Time-frequency analysis (Fourier transforms, short-time Fourier transform), transfer function
  - Filtering (low-pass, high-pass), sampling theorems (Nyquist)
  - Discrete-time systems, Z-transform, aliasing, quantization, DFT, FFT, DCT,
  - 2D extensions
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Describe** how sampling rate and aliasing affect digital audio or image acquisition.
    - *(Remember)* **Identify** key transforms (Fourier, Z-transform) used to analyze signals in the frequency domain.
  - **Methodological/Skills**
    - *(Apply)* **Implement** basic filtering in Python/MATLAB, demonstrating noise reduction on audio or image data.
    - *(Analyze)* **Examine** frequency spectra of signals (speech, image) and link them to AI tasks (speech recognition).
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** trade-offs in sampling rate and bit depth for sensor data in robotics or computer vision.
    - *(Create)* **Design** a small pipeline that processes raw sensor signals for subsequent ML classification (e.g., ECG signals).

AMCF-7: Cognitive Psychology & Neuroscience Fundamentals

- **Hours**: 30-45
- **Topics**:
  - Memory models (short-term, long-term, working memory), attention, language, affect
  - Basic neuroanatomy and CNS function (neurons, neurotransmitters, cortex, subcortical structures), neural imaging, sensorimotor integration
  - Theoretical links between biological intelligence and AI (connectionism, neural coding)
- **Learning Outcomes**:
  - **Content/Knowledge**

        - *(Understand)* **Explain** how human attention and memory models influence designs of attention-based AI.
        - *(Remember)* **Name** key brain regions relevant to perception, motion, and language.
    - **Methodological/Skills**
        - *(Analyze)* **Discuss** parallels between neural circuits and certain deep learning architectures.
        - *(Apply)* **Relate** cognitive constraints (e.g., working memory limits) to AI user-interface designs.
    - **Transferrable/Application**
        - *(Evaluate)* **Debate** ethical considerations of neuromorphic computing or brain–machine interfaces.
        - *(Create)* **Propose** ideas that adapt neuroscience findings (e.g., reinforcement signals in the brain) for advanced AI agents.

KA3: Core AI (CAI)

Description: The central AI courses. Each KU typically corresponds to **one semester** (or half a semester if combined with other units). Contains 8 KUs.

Recommended Total Hours for KA3: ~**300-420 contact hours** (depending on how deeply each subtopic is split into separate courses).

CAI-1: Foundations of Artificial Intelligence

- **Hours**: 30-45
- **Topics**:
    - Definition & history of AI, interdisciplinary connections of AI
    - Symbolic VS subsymbolic AI, AI paradigms:
        - Rules
        - Search
        - Learning
    - Problem-solving & search
    - Knowledge representation & reasoning, probabilistic reasoning & uncertainty
    - Learning from data
    - Intelligent agents (taxonomies, hierarchies, architectures):
        - Simple reflex, model-based, goal-based, utility-based, learning agents
        - Reactive, deliberative, hybrid agents

        - Single-layered, three-layered, subsumption, cognitive architectures
    - Narrow/Weak AI VS General/Strong AI
    - Philosophy of AI basics: Turing test, Chinese room, physical symbol system hypothesis, connectionism, etc.
    - Case studies of real-world AI systems (healthcare, robotics, finance, autonomous vehicles, recommendation systems, NLP)
    - AI ethics overview
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Compare** symbolic AI with subsymbolic AI in historical context.
        - *(Remember)* **Recall** major milestones (e.g., Turing test, expert systems, deep learning breakthroughs).
    - **Methodological/Skills**
        - *(Apply)* **Implement** a simple search-based solver for a puzzle (e.g., 8-puzzle), using simple tools (e.g., Jupyter Notebook).
        - *(Analyze)* **Debate** pros/cons of the Turing test as a measure of intelligence.
    - **Transferrable/Application**
        - *(Evaluate)* **Assess** high-level ethical concerns in early AI systems (e.g., medical expert systems).
        - *(Create)* **Formulate** a practical real-world scenario as a state-space search problem.

CAI-2: Knowledge Representation & Problem-Solving

- **Hours**: 30-45
- **Topics**:
    - Ontologies, description logics, semantic networks, knowledge graphs
    - Constraint satisfaction problems (CSPs), constraint propagation, backtracking
    - State space representation, BFS, DFS
    - Heuristic search & metaheuristics: admissibility, beam search, hill-climbing, A*, genetic algorithms, simulated annealing
    - Two-player adversarial games: minimax, alpha-beta pruning,
    - Semantic Web (OWL/RDFS)
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Describe** how ontologies and semantic networks structure domain knowledge.
        - *(Remember)* **Identify** basic CSP algorithms (backtracking, forward-checking, arc consistency).
    - **Methodological/Skills**
        - *(Apply)* **Encode** domain knowledge using an ontology language (OWL) and query it with SPARQL.

        - *(Analyze)* **Select** a metaheuristic (GA VS hill climbing) for large-scale optimization tasks.
    - **Transferrable/Application**
        - *(Evaluate)* **Collaborate** with domain experts to design an effective knowledge representation for a real scenario (e.g., healthcare).
        - *(Create)* **Implement** a CSP-based solver for scheduling or resource allocation, analyzing performance trade-offs.

CAI-3: Reasoning & Planning

- **Hours**: 30-45
- **Topics**:
    - Logic formalisms (propositional, first-order), inference engines, expert systems
    - Default and non-monotonic reasoning, belief revision, argumentation, resolution-based theorem proving
    - Model-based and case-based reasoning
    - Logic programming (PROLOG, answer set programming)
    - Probabilistic reasoning (Bayesian networks, Markov Decision Processes, fuzzy logic), Monte Carlo methods
    - Estimation theory, Hidden Markov Models, Bayesian filtering
    - Automated planning & scheduling (STRIPS, PDDL, GraphPlan)
    - Multi-agent decision-making, game theory basics
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** differences between deterministic logic-based reasoning and Bayesian/probabilistic approaches.
        - *(Remember)* **Name** common planning formalisms (STRIPS, HTN, PDDL).
    - **Methodological/Skills**
        - *(Apply)* **Encode** a planning domain and solve it with a classical planner (e.g., GraphPlan).
        - *(Analyze)* **Compare** single-agent MDP solutions vs. multi-agent approaches for uncertain environments.
    - **Transferrable/Application**
        - *(Evaluate)* **Assess** the performance of a planning system in partially observable settings (POMDP).
        - *(Create)* **Develop** a simple multi-agent simulation with negotiations or auctions, analyzing agent strategies.

CAI-4: Machine Learning

- **Hours**: 45-60
- **Topics**:
    - Statistical learning foundations: estimators, density estimation, bias-variance, overfitting, risk minimization, regularization

- Computational learning theory basics
- Supervised learning, unsupervised learning
- Model validation/selection, feature extraction/selection, ensemble methods (bagging, boosting, stacking)
- Linear models for classification and regression
- Kernel machines
- Non-parametric learning, curse of dimensionality, non-parametric algorithms
- Probabilistic models: Naïve Bayes classifiers, Gaussian Mixture Models, Expectation–Maximization algorithm (EM)
- Basic clustering algorithms: K-means/K-means++, DBSCAN, hierarchical clustering, spectral clustering
- Basic dimensionality reduction and manifold learning methods (e.g., LDA, PCA, ISOMAP, LLE)
- Dictionary learning, matrix factorization methods, Factorization Machines algorithm
- Association rule mining

- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** the bias-variance trade-off and how regularization counters overfitting.
    - *(Remember)* **Distinguish** supervised VS unsupervised paradigms with examples.
  - **Methodological/Skills**
    - *(Apply)* **Train** a classification model (e.g., logistic regression, decision tree) on real datasets, measuring performance.
    - *(Analyze)* **Perform** cross-validation, hyperparameter tuning, and interpret results for improvement.
  - **Transferrable/Application**
    - *(Evaluate)* **Discuss** ethical pitfalls (bias, privacy) in dataset curation and model deployment.
    - *(Create)* **Design** an ML pipeline from data preprocessing to final model integration in a small application.

CAI-5: Deep Learning

- **Hours**: 45-60
- **Topics**:
  - Perceptron, feed-forward ANNs, MLP and back-propagation
  - Universal approximation, autoencoders and representation learning
  - RBF networks, SOMs, Hopfield networks
  - Convolutional Neural Networks (CNNs)
  - Recurrent Neural Networks (RNNs), LSTMs
  - Attention and Transformers
  - Graph Neural Networks

- Self-supervised learning, weakly supervised learning, Foundation Models, multimodal AI
    - Training complex neural networks: batching, GPU acceleration, distributed training
    - Green AI practices, energy & carbon-footprint estimation
    - Neurosymbolic AI
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how CNNs handle spatial features differently from RNNs handling sequential data.
        - *(Remember)* **List** the main components of a Transformer (multi-head attention, feed-forward layers)
    - **Methodological/Skills**
        - *(Apply)* **Implement** a CNN or Transformer using a major library (TensorFlow/PyTorch).
        - *(Analyze)* **Examine** trade-offs (training speed, memory usage, performance) between different deep neural architectures.
    - **Transferrable/Application**
        - *(Evaluate)* **Compare** alternative approaches (e.g., MLP and LSTM, or CNN and Transformer) for recognition tasks.
        - *(Create)* **Fine-tune** a pretrained Foundation Model on a domain-specific dataset.

CAI-6: Computer Vision

- **Hours**: 45-60
- **Topics**:
    - Visual sensors, digitization, image formats
    - Projective geometry, image acquisition, image formation
    - 2D shape representation, image filtering, image segmentation
    - Edge/blob detection, feature extraction/detection/description, feature matching, image registration
    - Video capturing, video formats, video motion estimation, optical flow
    - Camera calibration, epipolar geometry, stereoscopic/3D/Multiview imaging, depth estimation
    - 3D reconstruction, Structure-from-Motion
    - Object recognition/detection/tracking, semantic segmentation
    - Scene understanding
    - Biometrics
    - Computational photography, hyperspectral imaging
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Summarize** how object recognition pipelines differ between using traditional handcrafted features and deep neural networks.

        - *(Remember)* **List** at least three image segmentation algorithms (thresholding, region-based, edge-based) and compare strengths/weaknesses.
    - **Methodological/Skills**
        - *(Apply)* **Implement** a basic feature extraction (corner detection) and classification pipeline on a known dataset.
        - *(Analyze)* **Compare** performance metrics (precision, recall, IoU) for object detection VS segmentation tasks.
    - **Transferrable/Application**
        - *(Evaluate)* **Train** a CNN for image classification, measuring accuracy against simpler methods that use handcrafted features.
        - *(Create)* **Develop** a small system than analyzes visual cues (e.g., from camera feed) for event recognition.

CAI-7: Natural Language Processing & Speech Analysis

- **Hours**: 45-60
- **Topics**:
    - Computational linguistics fundamentals, tokenization, normalization, stemming, lemmatization and Parts-of-Speech (POS) tagging
    - Text representations: Bag-of-Words, N-Grams, TF-IDF, static and contextual word embeddings
    - Syntax/semantics, language modeling (RNNs, Transformers)
    - Large Language Models
    - NLP tasks: question answering, text classification, document summarization, sentiment analysis, sentence similarity estimation, machine translation
    - Speech recognition/synthesis, acoustic modeling, speech segmentation
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** differences between different types of word representations (e.g., Word2Vec VS BERT).
        - *(Remember)* **Identify** main steps in text preprocessing.
    - **Methodological/Skills**
        - *(Apply)* **Implement** a text classification pipeline (e.g., sentiment analysis) using a standard library (spaCy, NLTK).
        - *(Analyze)* **Assess** speech recognition performance (WER) using open-source toolkits (Kaldi, DeepSpeech).
    - **Transferrable/Application**
        - *(Evaluate)* **Discuss** potential biases or fairness concerns in large-scale language models (ChatGPT-like).
        - *(Create)* **Deploy** a transformer-based QA system on a domain-specific corpus.

CAI-8: AI Ethics & Governance

- **Hours**: 30-45
- **Topics**:
    - Responsible AI, Trustworthy AI, value alignment
    - AI autonomy, personhood, safety, liability and security
    - Dual Use, legal frameworks (EU AI Act, GDPR, liability, data governance)
    - Global policy initiatives, AI ethics assessment tools
    - Ethical case studies: autonomous vehicles, facial recognition, deepfakes, fake news
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how laws and regulations (e.g., GDPR, AI Act) influence AI system design.
        - *(Remember)* **Identify** primary ethical dimensions (bias, privacy, transparency) in AI.
    - **Methodological/Skills**
        - *(Apply)* **Use** standard checklists or frameworks (model cards, datasheets for datasets) to document AI solutions.
        - *(Analyze)* **Audit** a dataset or trained neural model for potential biases or discriminatory outcomes.
    - **Transferrable/Application**
        - *(Evaluate)* **Debate** real-world controversies (e.g., predictive policing, face recognition) from multiple viewpoints.
        - *(Create)* **Propose** an ethical governance strategy for deploying AI in a high-stakes environment (e.g., medical, finance).

KA4: Advanced & Specialized AI Electives (ASAI)

Description: Additional topics that an AI department may offer as electives. Each KU typically has **20-45** recommended hours. Students select a subset based on their interests or departmental focus. Contains 15 KUs.

Possible Total Hours: **~200-300 contact hours per student** (depending on number of elective courses selectable by each student from this KA).

Institutions can convert any of these into **one or more courses** or half-semester modules.

ASAI-1: Reinforcement Learning & Sequential Decision Making

- **Hours**: 30-45

- **Topics**:
    - Markov Decision Processes with finite and discounted horizon
    - Bandits, exploration VS exploitation (ε-greedy, UCB)
    - Multi-armed bandits
    - Model-based VS model-free RL algorithms: Q-learning, SARSA, policy gradients
    - Deep RL: DQN, Rainbow, A2C, PPO
    - Multi-agent RL, hierarchical RL
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how exploration-exploitation trade-offs affect bandit and RL problems.
        - *(Remember)* **Identify** use-cases for different DRL algorithms (e.g., continuous VS discrete action spaces, etc.).
    - **Methodological/Skills**
        - *(Apply)* **Implement** Q-learning in a grid-world simulation, analyzing convergence speed.
        - *(Analyze)* **Compare** DRL methods (DQN VS policy gradient) on a benchmark environment (e.g., CartPole).
    - **Transferrable/Application**
        - *(Evaluate)* **Train** competing DRL agents in a simulator (e.g., intelligent video game bots), using different DRL algorithms, and compare their performance.
        - *(Create)* **Design** a multi-agent RL environment for resource-sharing tasks, measuring synergy and competition.

ASAI-2: Generative Artificial Intelligence

- **Hours**: 30-45
- **Topics**:
    - Generative models: Boltzmann machines, VAEs, GANs, Diffusion Models
    - Conditioning, mode collapse and mitigation measures
    - LLMs and Retrieval-Augmented Generation (RAG)
    - Prompt engineering and prompt learning for text, image, code generation
    - Metrics for generative quality (FID, IS, perplexity, hallucination rate)
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how different classes of generative models learn the underlying data distribution.
        - *(Remember)* **List** the major classes of generative models and their key differences.
    - **Methodological/Skills**

- *(Apply)* **Train** a GAN on a small image dataset, evaluating results with FID or IS.
- *(Analyze)* **Compare** Diffusion Models VS GANs for style transfer tasks, considering training stability.
- **Transferrable/Application**
  - *(Evaluate)* **Discuss** ethical issues around deepfake and synthetic media generation.
  - *(Create)* **Design** a generative AI system tailored to a specific application domain, selecting an appropriate model architecture and justifying design choices.

ASAI-3: Foundations of Trustworthy AI

- **Hours**: 20-30
- **Topics**:
  - Safety, robustness, adversarial training/attacks/defense
  - Fairness, bias mitigation, causal analysis
  - Accountability, reproducibility
  - Privacy & data governance, human oversight
  - AI alignment, RLHF
  - LLM security (prompt injection, jailbreaks)
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** multiple fairness metrics (e.g., statistical parity, equalized odds) and trade-offs.
    - *(Remember)* **Identify** fundamental sources of model vulnerability (e.g., adversarial examples, data poisoning, model inversion) and the key terms (robustness, interpretability, accountability) in trustworthy AI.
  - **Methodological/Skills**
    - *(Apply)* **Implement** adversarial training or reweighting to mitigate bias in a classification model.
    - *(Analyze)* **Examine** a trained AI system's predictions to detect fairness or safety issues, and propose evidence-based remedies.
  - **Transferrable/Application**
    - *(Evaluate)* **Propose** design changes that enhance reliability and trust for a real system (e.g., AI in hiring).
    - *(Create)* **Develop** a comprehensive policy framework or toolkit (documentation, checklists, monitoring protocols) that addresses fairness, privacy, and accountability from data collection through model deployment.

ASAI-4: Explainable AI (XAI)

- **Hours**: 20-30
- **Topics**:

    - Local VS global interpretability (LIME, SHAP)
    - Model-agnostic VS model-specific explanation
    - Post-hoc VS inherently interpretable models
    - Causal interpretability, counterfactual explanations
    - Visualization for interpretability (e.g., feature attributions, partial dependence plots)
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Describe** the difference between local (instance-level) and global interpretability.
        - *(Remember)* **Recall** key definitions, foundational concepts, and historical evolution of XAI.
    - **Methodological/Skills**
        - *(Apply)* **Use** LIME or SHAP on a black-box model to generate instance-level explanations.
        - *(Analyze*) **Evaluate** the trade-off between model accuracy and explainability in a regulated domain.
    - **Transferrable/Application**
        - *(Evaluate)* **Assess** the suitability of various XAI methods (e.g., SHAP, counterfactual explanations) for different types of models and stakeholder needs.
        - *(Create)* **Design** an end-to-end XAI pipeline for a chosen application domain.

ASAI-5: Distributed AI Systems

- **Hours**: 30-45
- **Topics**:
    - Multi-agent systems, agent coordination, auctions, negotiation
    - Federated learning, distributed deep learning, parameter servers
    - Resource scheduling, fault tolerance, and performance trade-offs in large-scale AI
    - Security and privacy considerations in distributed AI systems (e.g., secure aggregation)
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** the concepts of multi-agent coordination and negotiation protocols.
        - *(Remember)* **Recall** fundamental distributed AI terminologies, architecture patterns, and the rationale for federated VS centralized AI.
    - **Methodological/Skills**
        - *(Apply)* **Implement** a federated learning prototype, analyzing communication overhead.

        - *(Analyze)* **Examine** scalability challenges, performance bottlenecks, and fault-tolerance mechanisms in large-scale AI systems.
    - **Transferrable/Application**
        - *(Evaluate)* **Assess** trade-offs among efficiency, cost, security, and reliability when designing or selecting distributed AI infrastructure for real-world case studies.
        - *(Create)* **Develop** a multi-agent scenario (e.g., swarm robotics) with distributed AI decision-making.

ASAI-6: Human-Centered Machine Learning

- **Hours**: 30-45
- **Topics**:
    - Human-in-the-loop learning, interactive ML, user feedback loops
    - Collaborative/active learning, dialogue systems, user experience design, qualitative evaluation
    - Social/psychological aspects of ML adoption
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Describe** how user feedback can guide iterative model refinement (active learning).
        - *(Remember)* **List** key challenges in aligning ML models with human values, cognitive biases, and usability constraints.
    - **Methodological/Skills**
        - *(Apply)* **Prototype** an interactive labeling interface that updates ML models in real time.
        - *(Analyze)* **Compare** human-in-the-loop strategies (e.g., active learning, co-learning, interactive explanations) in terms of efficiency, cognitive load, and outcome quality.
    - **Transferrable/Application**
        - *(Evaluate)* **Assess** user satisfaction and trust in a human-centered ML system (UX metrics).
        - *(Create)* **Implement** a small-scale interactive ML application incorporating real-time human feedback and iterative refinement.

ASAI-7: AI for Music / Sound Analysis

- **Hours**: 20-30
- **Topics**:
    - Digital audio representation and processing (sampling, frequency domain), spectral analysis (STFT, MFCC)
    - Audio source separation, acoustic scene classification, sound event detection, audio representation learning
    - Speech/music transcription, genre/instrument classification

    - Music tagging/recommendation, music indexing and retrieval, similarity estimation, tempo estimation
    - Music generation (RNNs, Transformers)
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how spectral features (MFCCs, chroma) differ for speech VS music tasks.
        - *(Remember)* **Identify** common digital audio features (e.g., STFT, zero-crossing rate, spectral centroid) used in ML-based sound analysis.
    - **Methodological/Skills**
        - *(Apply)* **Develop** a music genre classifier, evaluating accuracy on a known dataset (GTZAN).
        - *(Analyze)* **Examine** failure cases in audio classification models, relating them to limitations in temporal resolution, noise, or dataset bias.
    - **Transferrable/Application**
        - *(Evaluate)* **Assess** the performance and limitations of a music tagging or recommendation system using precision, recall, and perceptual user studies.
        - *(Create)* **Implement** a music generation or style-transfer model, discussing creative/ethical impacts.

ASAI-8: Networked Intelligence

- **Hours**: 20-30
- **Topics**:
    - Complex network analysis, social network metrics and centrality measures (betweenness, closeness, eigenvector, etc.)
    - Link prediction, community detection, node classification, graph embeddings (GNNs)
    - Network information diffusion, random graph models
    - Recommender systems, content-based information retrieval
    - Blockchain, distributed consensus
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Describe** key centrality measures and their usage in social graphs.
        - *(Remember)* **Recall** fundamental network terminology, random graph models (e.g., Erdos-Rényi, Barabasi-Albert), and the basic properties that differentiate them.
    - **Methodological/Skills**
        - *(Apply)* **Use** a GNN library for node classification or link prediction.

        - *(Analyze)* **Analyze** real-world social network data, comparing multiple community detection or link-prediction algorithms and interpreting their differences in performance.
    - **Transferrable/Application**
        - *(Evaluate)* **Discuss** how network analysis can detect misinformation or malicious bot behavior on social platforms.
        - *(Create)* **Design** a small-scale end-to-end system that integrates network metrics into a real or simulated recommender or consensus-based application.

ASAI-9: Human-Centered Media Analysis

- **Hours**: 20-30
- **Topics**:
    - Face recognition/detection
    - Emotion analysis (facial expressions, text/speech sentiment), gesture/activity recognition, human body pose/posture estimation
    - Speaker recognition and diarization
    - Human-Computer Interaction (affective computing, bio-signals)
    - Privacy-enhancing methods (e.g., face de-identification)
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how emotional cues (facial, vocal) can be digitized and recognized by AI.
        - *(Remember)* **Identify** the key concepts and categories of human emotions (e.g., discrete VS dimensional models), along with the main relevant cues.
    - **Methodological/Skills**
        - *(Apply)* **Implement** a multimodal emotion recognition pipeline for short video clips.
        - *(Analyze)* **Analyze** the accuracy and biases of emotion recognition and gesture/activity detection systems across different demographic or cultural groups.
    - **Transferrable/Application**
        - *(Evaluate)* **Assess** the trade-offs between system performance and user privacy when deploying human-centered media analysis, including potential bias or misuse of sensitive data.
        - *(Create)* **Propose** privacy measures in a system that processes user videos.

ASAI-10: AI & Robotics

- **Hours**: 45-60
- **Topics**:
    - History and applications of robotics
    - Robot platforms (sensors, actuators), robotic paradigms
    - Kinematics, classical feedback (PID)

- Sensor fusion (EKF, UKF, Particle Filter, Visual-Inertial Odometry)
    - SLAM: EKF-SLAM, FastSLAM, visual SLAM
    - Motion planning: sampling-based, optimization-based
    - Optimal control: LQR, trajectory optimization
    - Imitation Learning & Behavior Cloning
    - Reinforcement Learning for robotics
    - Foundation Models for robotics
    - Sim-to-Real Transfer & Domain Randomization
    - Human-robot collaboration: shared autonomy, tele-operation, ergonomic & ethical aspects
    - Robotic programming & middleware
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Describe** differences between deliberative VS reactive robot control architectures.
        - *(Remember)* **List** the main sensors and actuators used in modern robotic platforms, their key characteristics and trade-offs.
    - **Methodological/Skills**
        - *(Apply)* **Implement** a basic SLAM approach in a simulator (ROS, Gazebo).
        - *(Analyze)* **Compare** sampling-based VS optimization-based motion planning algorithms.
    - **Transferrable/Application**
        - *(Evaluate)* **Discuss** safety and ethical considerations for autonomous vehicles or service robots.
        - *(Create)* **Develop** an end-to-end robotic application (ROS, Gazebo) that integrates perception, planning, and control to accomplish a specified task.

ASAI-11: AI & Games

- **Hours**: 20-30
- **Topics**:
    - Game engines
    - Classic AI for games: search algorithms, state machines, behavior trees, rule systems
    - Modern AI for games: reinforcement learning, neural networks, evolutionary computation
    - AI for Procedural Content Generation (PCG): grammar-based generation, cellular automata, generative learning
    - Player control, Non-Player Character control
    - Player modeling, multi-agent game AI
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how PCG fosters replayability in games.
        - *(Remember)* **Recall** the core search algorithms and state-machine structures used in game AI.

- **Methodological/Skills**
  - *(Apply)* **Implement** a player modeling system that adjusts difficulty based on user performance.
  - *(Analyze*) **Compare** the trade-offs between classic AI vs modern AI methods, in terms of adaptability, computational cost, and impact on player experience.
- **Transferrable/Application**
  - *(Evaluate)* **Assess** the effectiveness and fairness of a procedural content generator, identifying potential biases and mitigation strategies.
  - *(Create)* **Design** an RL-based NPC strategy for a real-time strategy (RTS) game environment.

ASAI-12: AI in Health Sciences

- **Hours**: 20-30
- **Topics**:
  - Medical imaging analysis (CT, MRI, X-ray), segmentation, classification
  - Genomics & proteomics, protein folding and structure prediction
  - Electronic Health Record (EHR) analysis, summarization and mining
  - Predictive modelling for disease risk
  - AI for drug repurposing and vaccine/drug design
  - Trustworthiness in clinical AI (e.g., privacy-preserving methods in federated settings, XAI, bias mitigation, etc.)
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Describe** how AI methods transform medical images, omics sequences, and EHR data into clinically useful insights (e.g., diagnosis, risk stratification, personalized treatment recommendations, etc.).
    - *(Remember)* **List** the primary AI architectures/methods and typical datasets/benchmarks used in health-sciences AI.
  - **Methodological/Skills**
    - *(Apply)* **Train** a medical image classification model, measuring sensitivity/specificity.
    - *(Analyze*) **Assess** model performance using clinical metrics (ROC AUC, sensitivity, specificity) and identify sources of bias or overfitting across demographic subgroups.
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** privacy, bias and ethical risks/compliance (HIPAA/GDPR) in a real telemedicine system.
    - *(Create)* **Design** a prototype AI-driven health application that integrates data preprocessing, model development, explainability tools, and validation steps.

ASAI-13: AI in Markets & Finance

- **Hours**: 20-30
- **Topics**:
  - Algorithmic trading, time-series forecasting, portfolio optimization
  - Risk management, fraud detection, credit scoring
  - Sentiment analysis for finance news, social media signals
  - Anti-Money Laundering (AML) algorithms, Know Your Customer (KYC) automation
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** fundamentals of time-series modeling (ARIMA, LSTM) for stock prices.
    - *(Remember)* **Recall** key portfolio-optimization methods (mean–variance, risk-parity) and standard risk metrics (VaR, CVaR, Sharpe ratio).
  - **Methodological/Skills**
    - *(Apply)* **Implement** anomaly detection for credit card fraud using clustering or supervised methods.
    - *(Analyze)* **Compare** the forecasting performance and interpretability trade-offs between ARIMA, LSTM, and Transformer models on historical price data using appropriate error and risk metrics.
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** the ethical, regulatory, and model-risk implications of AI systems in finance – critiquing AML/KYC automation, credit-scoring algorithms, and backtesting practices for fairness, transparency, and compliance.
    - *(Create)* **Design** a trading strategy backtest using real/historical data, analyzing risk metrics (VaR).

ASAI-14: AI in Social Science and Humanities

- **Hours**: 20-30
- **Topics**:
  - Computational social science, text mining of historical documents & archives
  - Digital humanities, text analysis for literature
  - Misinformation detection, fake data detection, Intellectual Property
  - Social activism, virtual communities
  - AI and justice, digital crime
  - AI in education
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Discuss** how AI-based text analysis aids historians or literary scholars (topic modeling).

        - *(Remember)* **Recall** core methods in computational social science and digital humanities (e.g., tf-idf, word embeddings, topic models, centrality metrics, and markup standards).
    - **Methodological/Skills**
        - *(Apply)* **Deploy** an NLP pipeline to mine named entities from archival texts.
        - *(Analyze)* **Compare** rule-based VS Transformer-based approaches for misinformation detection, evaluating precision, recall, and susceptibility to bias in historical or social-media corpora.
    - **Transferrable/Application**
        - *(Evaluate)* **Debate** the sociopolitical impact of AI-based misinformation detection.
        - *(Create)* **Implement** a digital-humanities prototype to explore a research question in social science or humanities (e.g., mapping ideological shifts in political speeches, reconstructing social networks from historical archives, etc.).

ASAI-15: Deep Arts

- **Hours**: 20-30
- **Topics**:
    - Image & sound restoration via AI
    - Generative AI for art, style transfer/restyling
    - Computational aesthetics, art curation
    - AI for narrative and storytelling
    - Generative AI and copyright
    - Immersive and interactive installations
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Explain** how style transfer methods transform paintings to new artistic styles.
        - *(Remember)* **Recall** the fundamental generative neural architectures and standard evaluation metrics (FID, IS for images; SNR, MOS for audio).
    - **Methodological/Skills**
        - *(Apply)* **Create** a generative model for image or music stylization.
        - *(Analyze)* **Compare** GANs, VAEs and Diffusion Models for image restoration and style transfer, evaluating trade-offs in image fidelity, diversity, computational cost, and artifact types.
    - **Transferrable/Application**
        - *(Evaluate)* **Discuss** intellectual property debates around AI-generated art in cultural contexts.
        - *(Create)* **Implement** an interactive AI art installation or narrative prototype that integrates a generative model, user interaction mechanisms, and deployment considerations.

KA5: AI Support & Professional Practice (AISP)

Description: Data engineering, project management, parallel/distributed programming, and innovation – supporting large-scale or commercial AI solutions. Contains 5 KUs.

Possible Total Hours: ~**90-150 contact hours per student** (depending on number of elective courses selectable by each student from this KA).

AISP-1: Data Management

- **Hours**: 30-45
- **Topics**:
  - Data preprocessing, cleaning & visualization
  - Data models (relational, NoSQL), query languages (SQL, SPARQL)
  - Data integration & transformation
  - Big Data, reliability, scalability, maintainability
  - Data warehousing, Extract-Transform-Load (ETL), OLAP
  - Apache Hadoop, Apache Spark
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** differences between relational and NoSQL data stores for AI.
    - *(Remember)* **Identify** ETL steps and typical pitfalls (inconsistent schemas, data cleaning).
  - **Methodological/Skills**
    - *(Apply)* **Design** a scalable data ingestion pipeline using a Big-Data framework (Spark).
    - *(Analyze)* **Assess** performance and fault-tolerance in distributed data processing.
  - **Transferrable/Application**
    - *(Evaluate)* **Discuss** privacy and compliance (e.g., GDPR) for personal or sensitive data in the pipeline.
    - *(Create)* **Implement** a Big-Data pipeline for large-scale machine learning training, measuring throughput.

AISP-2: Project Management

- **Hours**: 20-30
- **Topics**:
  - Requirements, specification, design, development, validation
  - Project lifecycles, Agile VS Waterfall model

- Scheduling (PERT, CPM), Gantt charts, critical path, slacks
  - Time/cost/risk management
  - Sustainability and team coordination
  - CI/CD & QA for ML pipelines, MLOps
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** differences between Agile and Waterfall models in iterative AI development.
    - *(Remember)* **Recall** key PM concepts (e.g., Gantt, WBS, risk register).
  - **Methodological/Skills**
    - *(Apply)* **Use** project management software (Jira, MS Project) to track tasks and deliverables.
    - *(Analyze)* **Mitigate** risks (e.g., data issues, model drift) throughout an AI project's life cycle.
  - **Transferrable/Application**
    - *(Evaluate)* **Assess** how sustainability, ethical and legal compliance considerations should be integrated into project scope.
    - *(Create)* **Plan** a semester-long AI capstone, detailing resources, schedules, and risk strategies.

AISP-3: Distributed Programming

- **Hours**: 20-30
- **Topics**:
  - MapReduce
  - Cloud computing models (IaaS, PaaS, SaaS), virtualization, containerization
  - Elasticity, scalability
  - Node configuration, orchestration (Kubernetes), cost evaluation
  - Data parallelism VS model parallelism in AI
- **Learning Outcomes**:
  - **Content/Knowledge**
    - *(Understand)* **Explain** how cloud service models differ for hosting AI microservices.
    - *(Remember)* **Recall** the differences between virtualization (VMs) and containerization (Docker), and define core concepts of elasticity, scalability, and orchestration in distributed systems.
  - **Methodological/Skills**
    - *(Apply)* **Deploy** a containerized AI service on a cloud platform, measuring latency and cost.
    - *(Analyze)* **Compare** data-parallel and model-parallel training approaches in terms of network overhead, compute utilization, and fault tolerance across multiple nodes.
  - **Transferrable/Application**

        - *(Evaluate)* **Compare** cost/performance trade-offs among different cloud configurations for large-scale model inference.
        - *(Create)* **Implement** a distributed AI pipeline that includes a MapReduce preprocessing job and a Kubernetes-orchestrated inference service with autoscaling and monitoring.

AISP-4: Parallel Programming

- **Hours**: 30-45
- **Topics**:
    - High-Performance Computing (HPC), race conditions, concurrency models: threads, message passing
    - MPI, OpenMP, CUDA, GPU/TPU programming, performance analysis
    - Scalability, speedup (Amdahl's Law), load balancing
- **Learning Outcomes**:
    - **Content/Knowledge**
        - *(Understand)* **Describe** concurrency issues (race conditions, deadlocks) in parallel code.
        - *(Remember)* **Recall** the main parallel-computing paradigms and list common synchronization primitives.
    - **Methodological/Skills**
        - *(Apply)* **Develop** GPU-accelerated code (CUDA kernels) for matrix operations used in deep learning.
        - *(Analyze)* **Assess** HPC cluster performance for training a complex Deep Neural Network, measuring scaling efficiency.
    - **Transferrable/Application**
        - *(Evaluate)* **Compare** the trade-offs between different parallelization frameworks (MPI, OpenMP, CUDA) in terms of programmability, performance, and resource utilization for a given workload.
        - *(Create)* **Implement** a heterogeneous parallel application with load balancing and performance optimizations, for a real-world scientific computing task.

AISP-5: Entrepreneurship & Innovation

- **Hours**: 20-30
- **Topics**:
    - Fundamentals of entrepreneurship, business model design (lean canvas), business plan
    - IP protection, data licensing
    - Market analysis, funding/investment, open innovation
    - Marketing and sales
- **Learning Outcomes**:
    - **Content/Knowledge**

- *(Understand)* **Explain** how intellectual property concerns apply to AI models/data.
- *(Remember)* **List** the nine building blocks of the Lean Canvas and the typical funding stages for tech start-ups.

- **Methodological/Skills**
  - *(Apply)* **Draft** a pitch deck for an AI startup, including monetization and go-to-market strategy.
  - *(Analyze)* **Deconstruct** a target market for an AI product: quantify TAM/SAM/SOM, map competitor value propositions, and identify strategic differentiation or "blue-ocean" niches.
- **Transferrable/Application**
  - *(Evaluate)* **Assess** the feasibility of a proposed AI product, considering ROI, regulatory constraints, and competition.
  - *(Create)* **Compile** a concise business plan that integrates financial projections, IP/data-licensing strategy, marketing & sales roadmap, and investment milestones suitable for presentation to potential investors.

Illustrative Competencies and Course Packaging

Similarly to CS2023, individual institutions can combine these Knowledge Units into actual courses, depending on needs. Examples:

- **"Programming and Data Structures"** (PFCS-1 & partial PFCS-4 in one 60-hour course)
- **"Mathematical Foundations for AI I"** (AMCF-1 & partial AMCF-2, ~60 hours)
- **"Mathematical Foundations for AI II"** (AMCF-3 & AMCF-5, ~60 hours)
- **"Foundations of AI"** (CAI-1 plus part of CAI-2, ~45 hours)
- **"Machine Learning and Deep Learning"** (CAI-4 & CAI-5 across two semesters, ~120 hours total)
- **"Reinforcement Learning"** (ASAI-1, single 45-hour elective)
- **"AI Ethics & Governance"** (CAI-8, single 30-45 hour course)

Sample Competency: *"Build and Deploy a Generative Model for Image Synthesis"*

- **Required KUs**:
  - AMCF-2 (Linear Algebra) & AMCF-3 (Probability) for understanding the math
  - CAI-5 (Deep Learning) or ASAI-2 (Generative AI) for building the model
- **Skill Level**: Develop, Optimize, Evaluate

Sample Competency: *"Implement a Multi-Agent System for Auction-Based Resource Allocation"*

- **Required KUs**:
  - PFCS-4 for data structures & efficiency
  - CAI-3 (Reasoning & Planning) for multi-agent decision-making
  - PFCS-5 (Computer Systems & Architecture)
  - ASAI-5 (Distributed AI Systems)
- **Skill Level**: Design, Code, Analyze Outcomes