Coincent
Learning Drives Excellence

Coincent 3 Year Program Curriculum
Artificial Intelligence

Partnered by

Microsoft

WORLD
RISK &
GOVERNANCE

Empowering Learners,
Accelerating Careers.

coincent.ai

ABOUT COINCENT

Coincent offers a 3-Year Program that is a well-structured, career-focused initiative designed to equip students with practical skills, real-world experience, and strong placement support. The program is tailored to ensure progressive learning and career readiness across three year phases.

Why It's Unique

- Only one batch per year with limited seats (150 students) per Domain to maintain quality.
- Prepares students step-by-step to become job-ready by graduation.

DETAILED ABOUT COINCENT 3 YEAR AI PROGRAM

“AI Program at Coincent – Learn by Doing”

Coincent’s AI program is designed to help students to build real-world skills through a structured, hands-on learning experience. With a blend of live sessions, recorded lectures, and guided projects, students gain deep insights into AI fundamentals,

machine learning, and neural networks — even without prior experience.

Here, mentors from top MNC’s will guide you and assist you through the sessions and live projects, training etc……

3-Year Program Structure Breakdown

TABLE OF CONTENTS

Year 1 :- Industrial Training

– "Before going to the curriculum preparation, Let's see the Basic Intro about AI and how coincent is going to deep dive into its chapters" –

What makes this program unique is

- it runs **parallel to your college academics**, allowing you to learn, apply, and grow without compromising your education.
- Through hands-on projects, Microsoft certifications, real-world mentorship, and a clear yearly progression — you'll be industry-ready by the time you graduate.

Program Overview

What Artificial Intelligence is and why it's important?

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think, learn, and make decisions. It enables computers and systems to perform tasks that typically require human intelligence, such as problem-solving, pattern recognition, and language understanding.

Importance of AI

- **Mimics Human Intelligence**

  AI systems are designed to imitate human cognitive functions like learning, reasoning, and problem-solving.

- **Learns from Data**

  AI uses large volumes of data to recognize patterns, make predictions, and improve its performance over time through techniques like machine learning.

- **Used in Everyday Applications**

  AI powers tools we use daily—such as voice assistants (e.g., Siri, Alexa), recommendation systems (e.g., Netflix, YouTube), and chatbots.

Chapter - 1 :Introduction to Artificial Intelligence

1.1 Python Overview:

Python is a high-level, interpreted programming language known for its simplicity and readability. It is widely used in various fields due to its versatility and ease of learning, making it a favorite among beginners and professionals alike.

Here are 4–5 key points about Python:

- Easy to Read and Write
- Versatile Language
- Large Community Support
- Cross-Platform Compatibility
- Rich Library Ecosystem
- Goal: Bring all students to a baseline in Python.
- Covers - `What is Python and why is it used in AI?

  Syntax basics: variables, loops, functions

  Writing a basic script (e.g., calculator or simple data handler)

1.2 Python Introduction and Installation:

Step-by-step instructions to:

- Download python(Windows, macOS, Linux)
- Set up IDEs (PyCharm, VS Code, or Jupyter Notebooks)
- Install packages using pip
- Check system path and virtual environments

1.3 Basics: Numbers, Strings

- Arithmetic operations (+, -, *, /, %, //)
- String Manipulation -

    Slicing, concatenation, formatting (f-strings)

    String methods (e.g., .lower(), .split(), .replace())

1.4 Data Types

- Primitive: int, float, str, bool
- Non-primitive: list, tuple, set, dict
- Type casting: int("5"), float(4)
- type() and isinstance() functions

Benefits and Outputs:

- Learn what makes Python ideal for AI, data science, and general-purpose programming.
- Set up Python and IDEs properly, manage packages and environments.
- Master core programming elements: variables, arithmetic, strings, and data types.
- Outcome: Gain confidence in writing basic scripts and working with different data formats.

Chapter 2: Introduction to Python Programming

2.1 Python Language Overview

- History of Python and its growing ecosystem
- Comparison with other languages (Java, C++)
- Syntax overview and use cases in data science and AI

2.2 Introduction to Python Programming

- Writing a complete program from scratch
- print(), input(), basic variables and logic
- Emphasis on code structure and readability

2.3 Basics of Python Programming

- If-else conditions
- Nested statements
- While and For loops
- Logic and comparison operators

2.4 Python Programming – Part 1

- Lists: creation, access, loop through
- Tuples: immutability and usage
- Dictionary: keys and values, looping
- List comprehensions and nested structures

2.5 Python Programming – Part 2

- Functions: def, arguments, return values
- Modules: creating and importing (math, random)
- File I/O: reading and writing text file

Benefits and Outputs:

- Understand Python's simplicity, readability, and relevance in modern tech fields.
- Build logical programs using loops, conditionals, and core data structures.
- Learn functions, modules, and file I/O for building modular, real-world applications.
- Outcome: Develop clean, efficient Python code suitable for data-driven projects.

Chapter 3: Introduction to Matplotlib (Data Visualization)

3.1 Introduction to Matplotlib

- Purpose: Visualizing trends and comparisons
- Introduction to pyplot, figure, axes

3.2 Matplotlib Plots

- Line, scatter, bar, histogram, pie charts
- Understanding x and y axes, legends, and grid

3.3 Matplotlib Styles – Part 1

Customizing:

- Title, x/y axis labels
- Font size, type, color
- Line styles (--, -, :)

3.4 Matplotlib Styles – Part 2

- Themes: seaborn styles, ggplot, bmh
- Using plt.style.use()
- Saving plots as images (.png, .svg)

3.5 Matplotlib Plot Variants

- Subplots: multiple plots in one figure
- Plotting from Pandas DataFrames
- Adding annotations, text, and markers

Benefits and Outputs:

- Create various plots to represent data trends and comparisons visually.
- Customize charts with titles, labels, styles, and annotations for better readability.
- Use themes, subplots, and export options for professional visual output.
- Outcome: Communicate insights effectively through clean, informative data visualizations.

Chapter 4: Introduction to NumPy (Numerical Computing)

4.1 Introduction to NumPy

- Why use NumPy over regular Python lists
- Efficiency and memory handling

4.2 NumPy Data Types and Operations

- Types: int32, float64, bool
- Element-wise operations
- Comparisons and logical operations

4.3 NumPy Arrays

- Creating arrays: np.array, np.zeros, np.arange, np.linspace
- Reshaping: .reshape(), .flatten()
- Indexing and slicing multidimensional arrays

4.4 Array Attributes

- .shape, .ndim, .dtype, .size
- Transpose: .T
- Copy vs View behavior

4.5 NumPy Math

- Basic: sum, mean, std, min, max
- Linear algebra: dot product, matrix multiplication
- Random number generation: np.random

Benefits and Outputs:

- Understand efficient data storage and computation using NumPy arrays.
- Perform element-wise operations, reshaping, slicing, and linear algebra tasks.
- Learn key array attributes and distinctions between views and copies.
- Outcome: Build a strong foundation in numerical computing for. data science and ML.

Chapter 5: Introduction to Pandas (Data Handling)

5.1 Introduction to Pandas

- Understanding Series and DataFrames
- Structure of tabular data

5.2 Reading Data

- read_csv(), read_excel(), read_json()
- Handling missing values: NaN, fillna(), dropna()

5.3 Data Manipulation – Part 1

- Selecting rows/columns using loc and iloc
- Adding/removing columns
- Filtering data using conditions

5.4 Data Manipulation – Part 2

- groupby(), aggregation
- Sorting by values or index
- Pivot tables and reshaping

5.5 Data Manipulation – Part 3

- Merging and joining DataFrames
- concat(), merge(), join()
- Time series basics (resample, rolling)

Benefits and Outputs:

- Understand and work with Series and DataFrames for structured data.
- Read, clean, and preprocess data using functions like read_csv(), fillna(), and dropna()
- Perform advanced data manipulation using groupby(), merge(), pivot tables, and time series methods.
- Outcome: Efficiently analyze and prepare data for insights and further ML workflows

Chapter 6: Machine Learning – Unsupervised Learning

6.1 Introduction to Machine Learning

- What is ML, and its use in real-world applications
- Types of learning: Supervised, Unsupervised, Reinforcement
- Algorithms: Clustering, Classification, Regression

6.2 Unsupervised Learning Techniques

- Clustering:
  - ➢ Visualizing clusters using PCA or t-SNE
  - ➢ K-Means, Hierarchical, DBSCAN
- Dimensionality Reduction:
  - ➢ PCA, LDA
  - ➢ Feature selection and extraction

Benefits and Outputs:

- Understand machine learning types and key algorithms like clustering, classification, and regression.
- Gain hands-on experience with unsupervised techniques such as K-Means, DBSCAN, and PCA.

- Learn to visualize and reduce high-dimensional data for better interpretation.
- Outcome: Ability to apply ML techniques to extract patterns and insights from real-world data.

Chapter 7: Introduction to TensorFlow and Deep Learning

7.0 Getting Started with TensorFlow

- Installing TensorFlow: Install using pip install tensorflow. It supports CPU and GPU versions.
- Tensors: Core data structure in TensorFlow—multi-dimensional arrays used in computations.
- Architecture: TensorFlow uses a computation graph; in v2.x, eager execution allows real-time operation.
- Basic Operations: Includes tensor creation, arithmetic (add, multiply), reshaping, and reduction (sum, mean).

7.1 Introduction to Keras

- Keras Overview: High-level API within TensorFlow for building neural networks easily.
- Building Models: Use Sequential or functional API to define layers; compile with optimizer, loss, and metrics.
- Training & Evaluation: Use fit() to train, evaluate() for test performance, and predict() for outputs.

.2 Clustering

- Model Building: Use Scikit-learn for classic clustering (K-Means, DBSCAN) or Keras with autoencoders.
- Visualization: Use PCA or t-SNE with matplotlib to plot clustered data in 2D.

- Evaluation: Silhouette Score measures how well data points fit within clusters (closer to 1 is better).

7.3 Deep Learning Basics

Understanding:

➢ Perceptrons, hidden layers, weights, biases

➢ Activation functions (ReLU, Sigmoid, Softmax)

Backpropagation and gradient descent

7.4 Natural Language Processing (NLP)

- Text cleaning (lowercase, punctuation removal)
- Tokenization, stemming, lemmatization
- Word embeddings: Word2Vec, TF-IDF
- Simple NLP tasks: Sentiment Analysis, Spam Detection

7.5 Reinforcement Learning

- Concepts: Agent, Environment, Reward, Policy
- Introduction to Q-Learning and Markov Decision Processes
- Real-life examples: games, robotics, recommendation systems

Benefits and Outcomes:

- Build and train deep learning models using TensorFlow and Keras.
- Apply clustering and visualization techniques for unsupervised learning.
- Perform basic NLP tasks like text cleaning and sentiment analysis.
- Understand reinforcement learning for applications in games and automation.

Year 2 :- Application & Project Phase:

– Year 2 is full of hands-on-experience on 8 live projects –

CAPSTONE PROJECTS

1. **News Classification using Natural Language Processing (NLP)** is a project that involves automatically categorizing news articles into predefined categories such as sports, politics, technology, or entertainment. It leverages NLP techniques to preprocess textual data—like tokenization, stop word removal, and stemming—followed by machine learning algorithms such as Naive Bayes or Support Vector Machines for classification. The model learns patterns from labeled training data to predict the category of new, unseen news articles. This project is widely used in media monitoring, personalized news feeds, and content organization.

2. **Text Classification with TensorFlow** is a process of automatically assigning categories to text data using deep learning models. Leveraging TensorFlow and Keras, you can build neural networks such as CNNs, RNNs, or LSTMs to learn patterns in text and classify it into labels like spam/ham, sentiment (positive/negative), or topic categories. The text is first preprocessed and converted into

numerical form using tokenization and embeddings before training the model. This technique is widely used in applications like sentiment analysis, email filtering, and content moderation.

3. **Handwritten Digit Classification** is a classic computer vision task where an AI model is trained to recognize digits (0–9) from images, typically using the MNIST dataset. Convolutional Neural Networks (CNNs) are commonly used in this project to automatically learn features like edges, shapes, and curves from the input images. The model is trained on thousands of labeled digit images and can then accurately classify new handwritten digits. This application is widely used in postal automation, bank cheque processing, and form digitization.

4. **Recognition of Objects** in AI involves detecting and classifying objects within images or video frames using deep learning techniques. Models
Convolutional Neural Networks (CNNs), YOLO (You Only Look Once), and Faster R-CNN are commonly used for identifying multiple objects with high accuracy and speed. Object recognition is crucial in applications such as autonomous vehicles, surveillance systems, and medical imaging. The process includes training models on large annotated datasets to learn

patterns and features specific to different object classes.

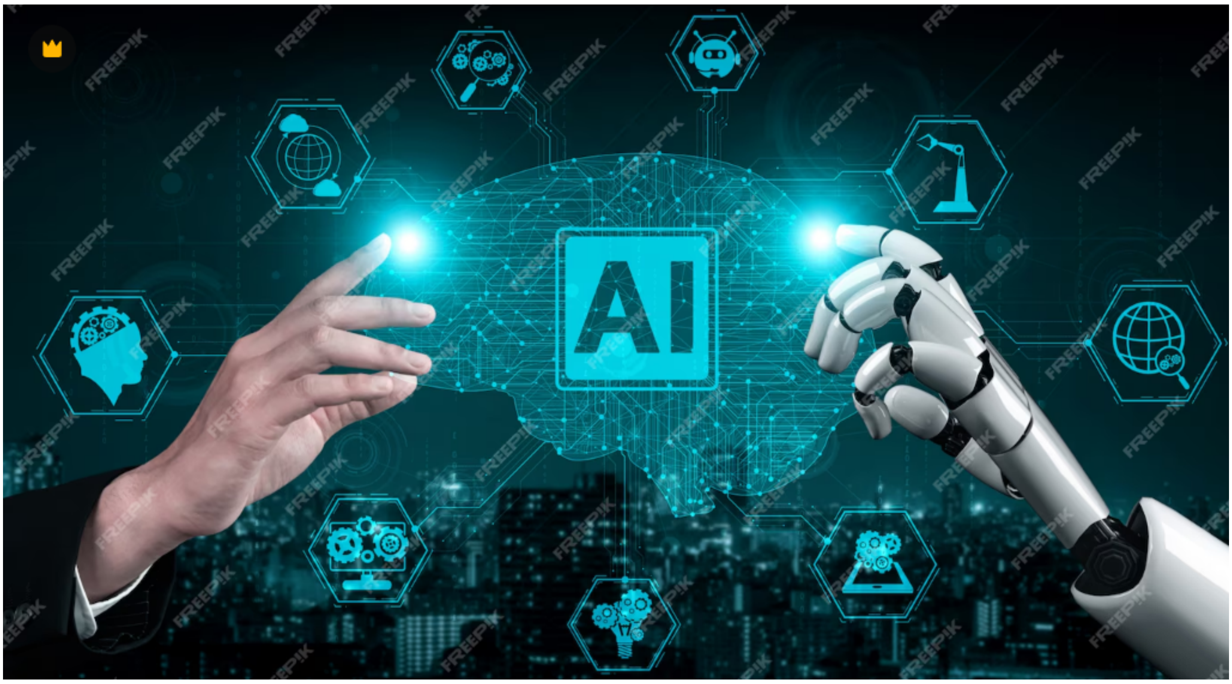
5. **Handwritten Digit Classification** is a classic computer vision task where an AI model is trained to recognize digits (0–9) from images, typically using the MNIST dataset. Convolutional Neural Networks (CNNs) are commonly used in this project to automatically learn features like edges, shapes, and curves from the input images. The model is trained on thousands of labeled digit images and can then accurately classify new handwritten digits. This application is widely used in postal automation, bank cheque processing, and form digitization.

6. **Automatic Speech Recognition (ASR)** is an AI technology that converts spoken language into written text. It uses deep learning models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), or Transformer-based architectures to analyze audio signals and transcribe speech accurately. ASR systems are used in voice assistants, transcription services, and real-time communication tools. The process involves feature extraction from audio, acoustic modeling, and decoding using language models.

7. **Landmark Detection** in AI involves identifying specific key points or features in images, such as the corners of the eyes, tip of the nose, or joints in the human body. It uses deep learning models like Convolutional Neural Networks (CNNs) to accurately locate these points, even in complex backgrounds or varying poses. This technique is widely used in facial recognition, gesture tracking, augmented reality, and medical imaging. The model is trained on annotated datasets to predict the coordinates of landmarks.

8. **Vision Transformer (ViT)** is an advanced deep learning model that applies the Transformer architecture, originally developed for natural language processing, to image recognition tasks. Unlike traditional CNNs, ViT divides images into fixed-size patches, flattens them, and treats them as a sequence of tokens—similar to words in a sentence. Each patch is embedded and passed through Transformer encoders that use self-attention mechanisms to capture global relationships. This allows ViT to learn long-range dependencies across an image more effectively than CNNs. Vision Transformers have shown competitive or superior performance on

benchmarks like ImageNet when trained on large datasets. They are particularly powerful in tasks requiring a global understanding of the visual scene. ViT models are scalable and benefit significantly from large-scale pretraining



Year 3 – Placement & Internship Phase:

1. Guaranteed Internship Phase

- In Year 3, Coincent guarantees an internship with partner companies for every student at no extra cost. The internship includes a formal Internship Offer Letter and a Completion Certificate upon successful completion.
- This is part of their "Industrial Training + Internship" model — training fees cover live classes, mentorship, and project work, but the internship phase itself is completely complimentary

2. Structured Placement Preparation

- Coincent supports students in portfolio-building with multiple completed projects (typically around 8+) and Microsoft-aligned certifications .
- They provide mock interviews, resume reviews, and training for HR and technical rounds — all aimed at preparing you for real-world hiring.

3. Final Take

- Coincent’s 3rd year transforms theory into practical experience through a guaranteed internship, builds a robust credentials portfolio, and equips you with placement-ready skills via mock interviews and resume prep. If you're in your 4th year, this phase sets you on a clear trajectory from "training" to "hired."

Step Into Top Tech Roles

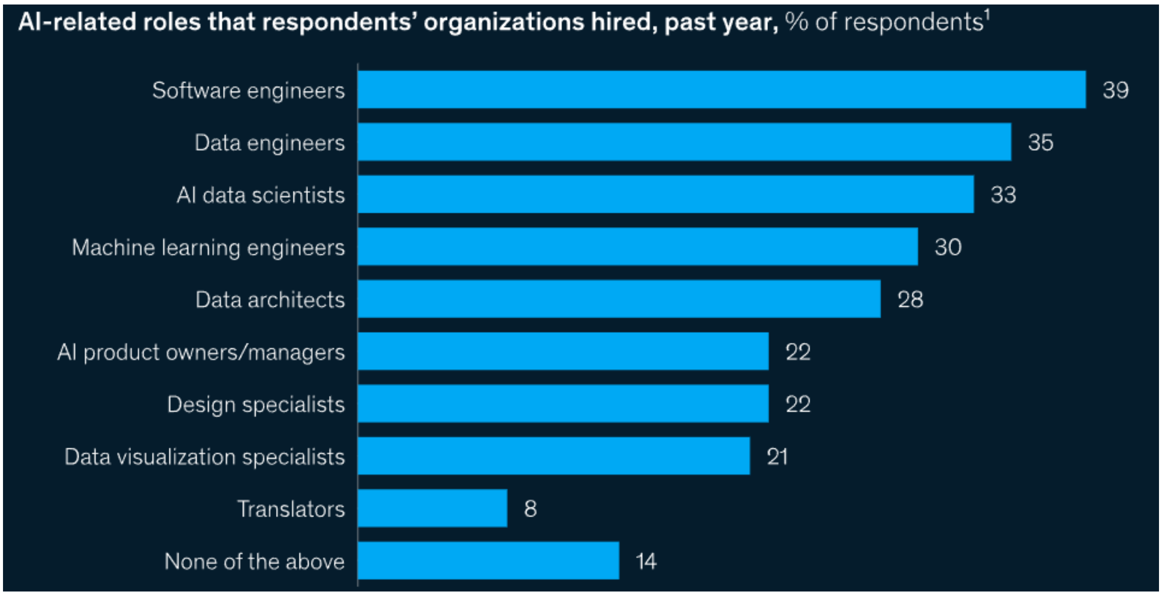
The top tech roles for someone skilled in artificial intelligence (AI) include a variety of specialized positions that leverage AI technologies across different industries. Some of the key roles include:

- AI Engineer: Designs and develops AI systems and algorithms, often requiring a strong background in math, science, engineering, and programming languages such as Python, C, and MATLAB.
- Research Scientist: Conducts advanced research in AI, often requiring a master’s or doctoral degree.

- AI Support Specialist: Helps users troubleshoot AI applications, providing technical support and assistance.

- Computer Vision Engineer: Focuses on extracting and analyzing information from visual data, such as images and videos, using tools like OpenCV and frameworks such as TensorFlow or PyTorch.

- Data Scientist: Analyzes large datasets to derive insights and inform decision-making, often using machine learning techniques



AI-related roles that respondents' organizations hired, past year, % of respondents[1]