

# PCEI™ Certified Entry-Level AI Specialist with Python

## Overview

The PCEI™ – Certified Entry-Level AI Specialist with Python is an internationally recognized certification issued by the Python Institute. It validates essential knowledge and foundational skills in Artificial Intelligence (AI), machine learning concepts, data handling, Python-based logic, neural networks, generative AI, and responsible AI practices. This certification course prepares participants for the PCEI-30-01 exam, covering all six exam blocks as defined in the official Python Institute syllabus (last updated December 2025). The course is designed for beginners with foundational Python skills, equipping them with the vocabulary, reasoning ability, and practical knowledge needed for entry-level AI roles.

## Who should enroll?

- The PCEI certification is designed for: Aspiring AI specialists seeking a foundational credential Retake Policy
- Students and career changers building job-ready AI knowledge
- Python developers wanting to move into AI and ML roles
- Professionals in non-technical roles who need to understand AI tools
- Educators and trainers introducing AI literacy in training programs
- Fresh graduates looking to differentiate themselves in the job market

## Prerequisites

PCEI is designed for beginners with foundational Python and data analysis skills — typically those who have completed PCEP™ (Certified Entry-Level Python Programmer) or equivalent. No prior AI experience is required. Comfortable with Python variables, loops, functions, and basic data handling Ability to read and write simple Python programs confidently Basic understanding of data structures (lists, dictionaries) No prior machine learning or AI experience required

## Course Outline

### Module 1: Artificial Intelligence Fundamentals (5)

#### 1.1 Define Artificial Intelligence and Its Core Concepts (1)

- Define Artificial Intelligence and related terminology (AI, agent, environment, inference);
- Distinguish between Narrow AI and General AI;
- Identify the components of an AI system (data, algorithms, conceptual models, the input → processing → output flow);
- Recognize examples of AI in everyday life (search, recommendations, assistants, navigation, image and text processing).

#### 1.2 Distinguish Major Subfields of AI (1)

- Describe the main subfields of AI (Machine Learning, Deep Learning, NLP, Computer Vision, Robotics);
- Match problems to appropriate subfields (e.g., text summarization → NLP);
- Describe how generative AI relates to these subfields.

#### 1.3 Describe How AI Systems Learn from Data (1)

- Explain the idea of learning from data using simple examples;
- Distinguish training from inference;
- Describe features, labels, and how simple models use them;
- Recognize the difference between labeled and unlabeled data in basic learning scenarios;
- Recognize feedback loops in AI systems;
- Identify how data quality affects learning (noise, bias, gaps).

#### 1.4 Identify the Capabilities and Limitations of AI Systems (1)

- Identify typical capabilities of AI systems (classification, prediction, pattern recognition);
- Recognize common limitations (lack of context, hallucinations, misclassification);
- Explain why AI performance depends on data quality and diversity;
- Distinguish situations where AI is appropriate vs. inappropriate.

#### 1.5 Identify Requirements for Building AI-Driven Solutions (1)

- Identify requirements needed to implement an AI solution (data availability, problem definition, objectives);

- Recognize success metrics for evaluating AI systems;
- Distinguish problems suitable for AI from those unsuitable.

## Module 2: Machine Learning Fundamentals (6)

### 2.1 Distinguish Types of Machine Learning (1)

- Describe the three primary learning types: supervised, unsupervised, and reinforcement learning;
- Match tasks to the appropriate learning type (e.g., classification → supervised, grouping customers → unsupervised);
- Recognize real-world examples of each learning type;
- Identify the components of a basic learning scenario such as agent, reward, and environment (reinforcement learning).

### 2.2 Describe the Machine Learning Workflow (1)

- Describe the steps in a machine learning workflow (data collection, cleaning, training, evaluation, inference);
- Explain the difference between training and testing datasets;
- Recognize the importance of clean, high-quality data;
- Identify common data issues (missing values, outliers, inconsistent types).

### 2.3 Identify Common Machine Learning Algorithms and Their Uses (1)

- Recognize common ML algorithm families (linear models, decision trees, k-nearest neighbors, clustering, Naive Bayes);
- Match basic algorithms to appropriate problem types (e.g., k-NN → classification; k-means → clustering);
- Understand at a high level how simple models make predictions (rules, distances, patterns).

### 2.4 Implement Simple Machine Learning Logic in Python (2)

- Implement simple classification logic using conditions and rules;
- Compute distances between data points using Python (e.g., Euclidean distance);
- Perform basic clustering or grouping of values in pure Python;
- Create simple ML-like programs without external frameworks (e.g., rule-based sentiment classifier, k-NN style classifier).

### 2.5 Evaluate Machine Learning Models Using Basic Metrics (1)

- Calculate accuracy from a set of predictions;

- Interpret precision, recall, and basic confusion matrices for small datasets;
- Identify false positives and false negatives using simple examples;
- Identify signs of overfitting and underfitting at a conceptual level;
- Recognize when a model is performing well or poorly.

## Module 3: Data Handling, Analysis, and Visualization (6)

### 3.1 Handle and Process Data Using Python (1)

- Read and write data from common file formats (CSV, text, JSON);
- Use Python data structures (lists, dictionaries, lists of dictionaries) to store and organize datasets;
- Access, modify, and extract values from nested structures;
- Identify and correct basic data inconsistencies (missing values, type mismatches, formatting errors);
- Convert between data types when preparing data (e.g., strings ↔ numbers).

### 3.2 Analyze Numerical and Categorical Data in Python (1)

- Calculate basic statistical measures such as mean, median, minimum, maximum, and frequency counts;
- Group and summarize small datasets using Python loops and comprehensions;
- Sort data by selected attributes (e.g., by score, by category);
- Identify patterns or trends in small datasets (e.g., most common category, highest/lowest values);
- Use the math module for simple numeric operations and formulas.

### 3.3 Work with Numerical Data and Distances for AI Tasks (1)

- Interpret lists of numbers as vectors suitable for simple ML-style operations;
- Calculate Euclidean and Manhattan distance, and other simple similarity measures using Python;
- Normalize numeric values within a range (e.g., 0–1) to make data comparable;
- Prepare basic feature sets for simple AI experiments (e.g., using height/weight pairs or text frequency counts);
- Recognize when normalization, scaling, or distance metrics are necessary in AI workflows.

### 3.4 Use Python Tools to Organize and Prepare Data for Machine Learning (1)

- Load data using basic functions and simple pandas operations;

- Filter rows or items based on conditions (e.g., select students with scores > 80);
- Select specific attributes or columns from datasets;
- Combine or restructure data (e.g., merging lists, flattening nested dictionaries);
- Recognize the difference between feature selection and feature extraction;
- Recognize the importance of data preparation before training a model.

### 3.5 Visualize Data Using Python (1)

- Create basic visualizations using Matplotlib (line charts, bar charts, histograms);
- Customize charts with titles, labels, legends, and simple styling;
- Interpret patterns shown in charts (trends, distribution shapes, outliers);
- Choose appropriate chart types for different kinds of data (e.g., histogram for distributions, line chart for trends);
- Use visualizations to support reasoning data quality and ML readiness.

### 3.6 Explain the Importance of Data Quality in AI (1)

- Identify common data problems (noise, missing values, duplicates, inconsistent formatting, labeling errors);
- Explain how poor data quality can lead to inaccurate or biased predictions;
- Describe the role of preprocessing in AI workflows;
- Recognize when a dataset is not suitable for training (too small, low diversity, incorrect or inconsistent labels);
- Explain the link between clean data and model performance in simple terms.

## Module 4: Neural Networks, Deep Learning, and Generative AI (8)

### 4.1 Describe the Foundations of Neural Networks

- Identify the components of a simple neural network (neurons, weights, biases, layers)
  - Explain the role of activation functions in simple terms (e.g., “deciding when a neuron fires”)
  - Describe how signals move through a network (feedforward process)
  - Explain the idea of backpropagation as adjusting weights to reduce mistakes
- Distinguish between shallow and deep networks at a high level

### 4.2 Distinguish Classical Machine Learning From Deep Learning (1)

- Describe how deep learning differs from classical ML techniques
- Identify tasks where deep learning is more suitable (e.g., image recognition, speech processing);
- Match sample problems to appropriate approaches (ML vs DL);

- Recognize the advantages and limitations of deep models (data hunger, computational cost)

#### 4.3 Describe the Fundamentals of Natural Language Processing (NLP) (1)

- Explain how computers represent text (tokens, sequences, embeddings)
- Recognize common NLP tasks (sentiment analysis, translation, summarization)
- Identify how rule-based and machine-learned approaches differ;
- Describe how context influences model outputs in language tasks. (data hunger, computational cost)

#### 4.4 Describe the Fundamentals of Computer Vision (CV) (1)

- Describe how images are represented as numerical arrays (pixels, color channels)
- Recognize common CV tasks (classification, detection, segmentation);
- Explain at a high level how convolutional neural networks process images;
- Identify everyday uses of CV (face recognition, scanning, autonomous vehicles)

#### 4.5 Explain the Principles of Generative AI and Large Language Models (1)

- Describe how images are represented as numerical arrays (pixels, color channels)
- Recognize common CV tasks (classification, detection, segmentation);
- Explain at a high level how convolutional neural networks process images;
- Identify everyday uses of CV (face recognition, scanning, autonomous vehicles)

#### 4.6 Describe the Fundamentals of Computer Vision (CV) (1)

- Describe how images are represented as numerical arrays (pixels, color channels)
- Recognize common CV tasks (classification, detection, segmentation);
- Explain at a high level how convolutional neural networks process images;
- Identify everyday uses of CV (face recognition, scanning, autonomous vehicles)

#### 4.7 Describe the Fundamentals of Computer Vision (CV) (1)

- Describe how images are represented as numerical arrays (pixels, color channels)
- Recognize common CV tasks (classification, detection, segmentation);
- Explain at a high level how convolutional neural networks process images;
- Identify everyday uses of CV (face recognition, scanning, autonomous vehicles)

## Module 5: Responsible AI, Ethics, and Critical Thinking (6)

### 5.1 Identify Ethical Risks and Safety Concerns in AI (1)

- Identify common ethical risks such as bias, discrimination, unfair outcomes, and harmful stereotypes;
- Recognize privacy concerns related to data collection, use, and storage;
- Describe safety issues such as hallucinations, misinformation, and inappropriate model outputs;
- Distinguish between safe and unsafe use of AI technologies in everyday scenarios.

### 5.2 Apply Basic Safety and Security Practices When Using AI Systems (2)

- Recognize which types of information should not be shared with AI systems (e.g., personal data, passwords, private documents);
- Identify unsafe or risky system interactions such as attempts to bypass controls, manipulate model behavior, or access restricted content;
- Follow basic security and safety guidelines when using AI tools, including data minimization, responsible input handling, and adherence to terms of use;
- Understand when to escalate or stop using an AI system if unexpected or potentially harmful behavior occurs.

### 5.3 Analyze the Social and Economic Impact of AI (1)

- Describe how AI affects society, education, industry, and public services;
- Identify ways AI transforms jobs and creates a need for reskilling;
- Recognize the risks of job displacement and widening inequalities (digital divide);
- Evaluate both positive and negative impacts of AI adoption.

### 5.4 Explain Principles of Responsible, Transparent, and Human-Centered AI (1)

- Explain the principles of responsible AI (fairness, transparency, accountability, explainability);
- Describe the concept of “human-in-the-loop” and why human oversight is required;
- Recognize the importance of verifying AI outputs, checking sources, and applying human judgment;
- Identify practices that promote safe and trustworthy AI deployment.

### 5.5 Apply Critical Thinking to Evaluate AI Outputs (1)

- Identify when AI-generated responses are incorrect, misleading, or inconsistent;
- Detect hallucinations, unsupported claims, and logical errors in AI outputs;
- Cross-check AI answers using reasoning and external knowledge;

- Determine when human decision-making should override automated outputs.

## Module 6: AI Projects, Collaboration, and Communication (5)

### 6.1 Identify Opportunities for AI Solutions in Real-World Contexts (1)

- Identify simple, realistic problems that can be supported by AI (classification, prediction, summarization, recommendation);
- Distinguish between problems that are appropriate for AI and those that require traditional programming or human judgment;
- Recognize project goals, inputs, outputs, and basic constraints when planning an AI solution;
- Describe how data availability, quality, and domain knowledge influence project feasibility.

### 6.2 Recognize Basic Cost Considerations When Planning AI Projects (1)

- Identify common factors that influence the cost of using AI systems, such as data volume, model complexity, computational requirements, and use of external AI services;
- Distinguish between free, low-cost, and paid AI tools or APIs, and understand how usage limits may affect project design;
- Estimate whether a small AI project is feasible within typical resource constraints (time, storage, compute, or budget);
- Recognize situations where simpler approaches (rule-based logic, smaller models, offline tools) may reduce cost without affecting project goals.

### 6.3 Apply Basic AI Development Steps to a Small Project (1)

- Gather or load a small dataset suitable for a simple AI task;
- Prepare and transform data (cleaning, selecting attributes, summarizing);
- Implement a simple solution using Python (rule-based classification, k-NN-style logic, basic predictive workflow);
- Evaluate the output of the solution using simple metrics (accuracy, counts, comparisons);
- Identify limitations and areas for improvement in the project.

### 6.4 Collaborate Effectively in AI Projects (1)

- Describe the roles commonly involved in AI projects (analyst, developer, domain expert, reviewer);

- Use basic collaboration practices such as version control, shared responsibilities, and peer feedback;
- Participate in pair programming, code reviews, or simple teamwork workflows;
- Recognize the importance of documenting decisions and maintaining reproducibility.

## 6.5 Communicate AI Findings Clearly to Technical and Non-Technical Audiences (1)

- Present AI project results using charts, summaries, and simple explanations;
- Explain model outputs, limitations, and errors in clear, accessible language;
- Write short technical summaries describing methods, results, and conclusions;
- Tailor communication style based on the target audience (e.g., classmates vs. non-technical stakeholders).