

Course Name : **PCED – CERTIFIED ENTRY-LEVEL DATA ANALYST WITH PYTHON**
Duration : 3 Days (Physical Classroom / Virtual Live Instructor)
Skill Level : Beginner



COURSE DESCRIPTION:

Build foundational data analysis skills with Python through the PCED – Certified Entry-Level Data Analyst with Python certification training. This course covers data fundamentals, Python programming, data preparation, basic analysis, exploratory data analysis (EDA), and communicating data-driven insights.

Participants will learn how to work with structured data, clean and transform datasets, perform basic descriptive statistics, use NumPy for numerical analysis, identify trends and outliers, and communicate findings through clear visualizations and reports.

WHAT WILL YOU LEARN?

In this course, participants will develop the foundational knowledge and practical skills required for the PCED-30-02 certification, including Python programming for data analysis, file and CSV handling, data cleaning, descriptive statistics, exploratory data analysis, basic visualization, data storytelling, and analytical reporting.

PREREQUISITE:

No formal prerequisites. Recommended knowledge includes PCEP or equivalent foundational Python knowledge, together with a foundation in mathematics and statistics.

METHODOLOGY:

This program will be conducted with interactive lectures, PowerPoint presentations, discussions, demonstrations, and practical exercises. This course can be conducted as instructor-led (ILT) or virtual instructor-led training (VILT).

JOB SCOPE:

Upon completion of this course, candidates may pursue the following career paths:

- Junior Data Analyst
- Entry-Level Data Analyst
- Data Analytics Assistant
- Junior Python Data Analyst
- Data Reporting Analyst
- Business Data Analyst – Entry Level
- Data Operations Analyst
- Reporting & Analytics Executive
- Junior Data Specialist

MODULE 1: INTRODUCTION TO DATA AND DATA ANALYSIS

Define Data and Explain How It Becomes Meaningful

- Define data and explain its role in decision-making, business, and everyday life.
- Distinguish between data, information, and knowledge.
- Explain how raw data is processed and interpreted to produce usable insights.

• Classify Data by Type and Format

- Identify quantitative and qualitative data.
- Differentiate structured, semi-structured, and unstructured data.

• Data Sources and Collection Methods

- Identify data sources such as APIs, web pages, databases, IoT devices, surveys, and logs.
- Understand surveys, interviews, observations, automated systems, and web scraping as collection methods.
- Understand representative sampling and the implications of biased or incomplete data.

• Data Storage and Organization

- Describe CSV, JSON, Excel, and database formats.
- Understand data lakes, data warehouses, relational databases, and metadata.

• Data Lifecycle and Management

- Understand collection, storage, processing, analysis, visualization/reporting, archiving, and deletion.
- Understand how errors or poor data management can affect results and decisions.

• Data Analysis, Analytics, and Data Science

- Differentiate data analysis, data analytics, and data science.
- Understand descriptive, diagnostic, predictive, and prescriptive analytics.

• Ethical and Legal Considerations

- Understand transparency, consent, privacy, fairness, and accountability.
- Recognize GDPR, HIPAA, and CCPA and the role of anonymization and encryption.

MODULE 2: PYTHON BASICS FOR DATA ANALYSIS

• Variables and Data Types

- Define and assign variables using the assignment operator.
- Work with integers, floating-point numbers, strings, and Boolean values.
- Perform basic numeric and string operations.
- Use `type()` and `isinstance()` to inspect variable types.

• Python Data Collections and Sequences

- Create, access, and modify lists using indexing and slicing.
- Use list methods and list comprehensions to manage, transform, and filter data.
- Work with tuples and understand tuple immutability.
- Create sets and perform union, intersection, difference, and membership operations.
- Use sets to remove duplicates.

• Dictionaries and Strings

- Create, access, update, and delete dictionary values using keys.
- Use `dict.get()`, dictionary loops, and lists of dictionaries for lookup, counting, categorization, and structured records.
- Work with strings using indexing, slicing, loops, and common string methods.

- **Functions and Exceptions**

- Define functions using def; use positional, keyword, and default parameters and return values.
- Understand None, pass, local and global variables, and name shadowing.
- Handle TypeError, ValueError, IndexError, and FileNotFoundError using try-except.

- **Control Program Flow**

- Use comparison and logical operators with Boolean expressions.
- Use if, elif, and else for conditional logic.
- Use for and while loops with break, continue, and loop else.
- Combine loops and conditions for data cleaning, aggregation, and transformation.

- **Modules and Packages**

- Import built-in modules using import, from ... import, and aliases.
- Use math, random, statistics, collections, os, datetime, and csv in data-related tasks.
- Understand built-in versus third-party packages and install external libraries such as NumPy using pip.
- Use official Python and NumPy documentation to learn functions and troubleshoot problems.

MODULE 3: WORKING WITH DATA AND PERFORMING SIMPLE ANALYSES (CONTINUED)

- **Read and Write CSV Data**

- Use csv.reader() to read structured data from CSV files.
- Use csv.writer() to write tabular data to CSV format.
- Parse and clean text using strip() and split(',') where appropriate.
- Produce formatted summaries using f-strings.

- **Clean and Prepare Data**

- Identify missing or null-like values such as None and empty strings.
- Replace or remove missing values using logical checks.
- Detect invalid types, unexpected formats, and out-of-range values.
- Remove duplicates using sets, dictionary keys, or comprehensions.
- Apply simple min-max normalization.
- Track indexes with enumerate() when transforming data.

- **Clean and Format Strings and Data Types**

- Normalize text using strip(), lower(), upper(), replace(), and title().
- Chain string operations for multi-step cleaning.
- Convert values using int(), float(), str(), and bool().
- Format numeric values using f-strings.
- Use split() and join() for string fields.
- Parse and format dates and times using datetime.strptime() and strftime().

- **Perform Basic Analytical Computations**

- Use len(), sum(), min(), max(), and round() for basic aggregations.
- Count values using count() and dictionary accumulation patterns.
- Calculate mean, median, and standard deviation with the statistics module.
- Use math.sqrt(), math.ceil(), and math.floor() for basic numeric calculations.
- Use collections.Counter() for categorical frequency counts.

- **Numerical Analysis with NumPy**

- Convert lists to NumPy arrays using numpy.array().
- Use numpy.mean(), numpy.median(), numpy.std(), and numpy.sum().
- Generate sequences using numpy.arange() and numpy.linspace().

- **Conditional Metrics and Group Summaries**

- Calculate metrics for filtered subsets of data.
- Group values by categories and calculate summaries using dictionaries or loops.
- Combine multiple conditions using and/or for specific filters.

- **Exploratory Data Analysis (EDA)**

- Sort and filter data using sorted(), numpy.sort(), filter(), comprehensions, and logical conditions.
- Identify unique values using set() and numpy.unique().
- Count frequencies using Counter().
- Calculate basic correlations using numpy.corrcoef().
- Detect and filter simple outliers using thresholds, standard deviation, Boolean indexing, or conditions.

MODULE 4: COMMUNICATING INSIGHTS AND REPORTING

- **Basic Data Visualization**

- Identify bar charts, line charts, and pie charts and understand their purposes.
- Discuss strengths and limitations of common visualization types.
- Select appropriate visuals based on data type and communication goals.
- Interpret trends, comparisons, and proportions shown in simple visualizations.
- Identify misleading or unclear visuals and explain how they can be improved.

- **Data Storytelling**

- Structure a data story using introduction, insights, and conclusion.
- Lead with a key message supported by evidence.
- Use transitions and signposting to create a clear flow.
- Adjust tone, language, and depth for the intended audience.

- **Analytical Reporting**

- Summarize key patterns and findings using supporting data.
- Organize reports using problem, analysis, insight, and recommendation.
- Use headings, bullet points, and visuals to improve clarity and readability.

- **Communicating Insights in Presentations**

- Apply accessible and clean design principles to data presentations.
- Explain charts and analytical results clearly.
- Respond to questions using evidence from visuals or numerical findings.

- **PCED-30-02 Exam Preparation**

- Review the four PCED examination blocks and their key competencies.
- Practice applying Python fundamentals to data-related tasks.
- Review data cleaning, basic statistics, NumPy, filtering, grouping, EDA, and visualization concepts.
- Practice interpreting data and communicating evidence-based insights.

- **EXAM STRUCTURE**

- Block 1: Introduction to Data and Data Analysis Concepts – 9 items (22.5%).
- Block 2: Python Basics for Data Analysis – 13 items (32.5%).
- Block 3: Working with Data and Performing Simple Analyses – 13 items (32.5%).
- Block 4: Communicating Insights and Reporting – 5 items (12.5%).
- Total: 40 single-select and multiple-select items.
- Passing requirement: cumulative average score of at least 75% across all exam blocks.