Infotek
Solutions

Why This Course is a Game-Changer

In today's data-driven world, companies aren't just looking for "data people" — they're looking for problem solvers who can turn raw numbers into actionable insights. This course arms you with Python programming foundations and the industry's most in-demand tools — Pandas, NumPy, and Matplotlib — to make you job-ready fast.

Who Should Join?

- Students or graduates looking to break into **data careers**.
- Manual testers, business analysts, or software engineers looking to **upskill in data**.
- Anyone curious about **Python, analytics, and data visualization**.

Roles You Can Target After This Course

- Data Analyst
- Business Analyst (Data-Focused)
- Reporting Analyst
- Data Visualization Specialist
- Python Developer – Data Tools
- Junior Data Scientist
- MIS Analyst
- Operations/Data Reporting Executive

Python Programming

- **Python Introduction**
    - What is Python?
    - How Python works
- **Installation and setup**
    - Download and install python
    - PyCharm installation and setup
- **Basics of Python**
    - Print statement in Python
    - Print hello world in Python

  - Identifiers in Python
  - input() function in Python
- **Data Types in Python**
  - Different types of data types in Python
  - Data types code in Python
  - Type casting in Python
- **Operators in Python**
  - Arithmetic operator in Python
  - Relational operators in Python
  - Unary operators in Python
- **Control flow**
  - Control statement in Python
  - If-else in Python ○ Nested if-else in Python
  - Switch case in Python
  - Looping statements in Python
- **Functions**
  - Local and Global Variables
  - Defining and Calling Functions in Python
  - Arguments and Return Values
- **List in Python**
  - List in Python
  - List Methods and Operation in Python
  - List Example Code in Python
- **Tuple in Python**
  - Tuple in Python
  - Tuple Example Code in Python
- **Set in Python**
  - Set in Python
  - Set Example Code in Python
- **Dictionaries in Python**
  - Dictionaries in Python
  - Dictionaries Example in Python

- **OOPs (for Testing)**
  - Class and Object
  - Introduction of OOPs concept
  - Class and Object in Python
  - Initialization of Class and Object
  - OOPs Concept features
  - Inheritance in Python
  - Method Overloading and Method overriding
  - Polymorphism in Python

    - Data Abstraction in Python
    - Encapsulation in Python
- **List Comprehensions**
    - List Comprehensions in python
    - Recursion in Python
    - Recursion Scenario Example in python

- **Advanced Functions**
    - Lambda Functions in python
    - _args and _kwargs in Python
    - _args and _kwargs Example in Python
    - Decorators in Python
    - Decorator Example in Python
    - Generators in Python
    - Generators Scenario Example in python

- **Advanced Data Structure**
    - Stacks in Python
    - Stacks Scenario Example in python
    - Queue in Python
    - Queue Scenario Example in python
    - LinkedList in Python
    - LinkedList Scenario Example in python

- **Regular Expression**
    - RegularExpression in Python

- **File Handling in Python**
    - File Handling in Python
    - File Handling Scenario Example in python

- **Exception Handling**
    - Exception Handling in Python
    - Exception Handling Scenario Example in python

Section 1: Pandas – Data Manipulation & Analysis

1. **Introduction to Pandas**
    - What is Panda
    - Installing Pandas
    - Series vs DataFrame
2. **Data Input & Output**
    - Reading CSV, Excel, JSON, and SQL files
    - Writing data to CSV and Excel
    - Reading from URLs

3. **Data Exploration**
    - head(), .tail(), .info(), .describe()
    - Data types and conversion
4. **Data Selection & Indexing**
    - Selecting rows and columns with loc[] and iloc[]
    - Boolean indexing
    - Slicing DataFrames
5. **Data Cleaning**
    - Handling missing data: .dropna(),.fillna()
    - Replacing values: .replace()
    - Removing duplicates: .drop_duplicates()
6. **Data Transformation**
    - Renaming columns
    - Applying functions with .apply() and .map()
    - String operations in Pandas

7. **Sorting & Grouping**
    - Sorting data: .sort_values(), .sort_index()
    - Grouping and aggregations: .groupby(), .agg()

8. **Merging & Joining**
    - Concatenation (pd.concat)
    - Merge & Join (.merge(), .join())

9. **Time Series with Pandas**
    - Date parsing and formatting
    - Resampling and shifting

Section 2: NumPy – Numerical Computing

1. **Introduction to NumPy**
    - Why NumPy is fast (understanding ndarray)
    - Installing NumPy
2. **Creating Arrays**
    - From lists and tuples
    - Using np.zeros(), np.ones(), np.arange(), np.linspace()
    - Random arrays with np.random

3. **Array Operations**
    - Arithmetic operations
    - Broadcasting
    - Comparison operations
4. **Indexing, Slicing & Reshaping**
    - Indexing arrays
    - Slicing subarrays
    - Reshaping with .reshape() and .ravel()

5. **Statistical & Mathematical Functions**
    - np.mean(), np.median(), np.std()
    - np.sum(), np.min(), np.max()
    - Trigonometric functions

6. **Advanced NumPy**
    - Copy vs View
    - Stacking arrays (np.vstack, np.hstack)
    - Splitting arrays

Section 3: Matplotlib – Data Visualization

1. **Introduction to Matplotlib**
   - Installing Matplotlib
   - Figure, Axes, and Plot basics

2. **Line Plots**
   - Customizing lines (color, style, width)
   - Adding labels and legends
3. **Bar Charts**
   - Vertical & horizontal bars
   - Stacked bar charts
4. **Scatter Plots**
   - Plotting relationships between variables
   - Adding size and color dimensions

5. **Histograms**
   - Plotting frequency distributions
   - Bin customization

6. **Pie Charts**
   - Creating pie charts
   - Exploding slices

7. **Customizing Plots**
   - Titles, labels, grid, annotation
   - Color maps
8. **Multiple Plots**
   - plt.subplot() & plt.subplots()
   - Shared axes