ELE 503

Advanced Computer Programming and Statistics

Week#9: Python Programming for Engineers

By Kingsley E. Erhabor

Week 09.

Python Programming for Engineers

Python syntax and fundamental programming

Principles and Methodologies

Python libraries

Understanding the Role of Python Libraries

Python Real World Examples

Applications in Engineering

Best Practices in Python Programming

Understanding Python Best Practices

Q&A

Closing Take away

Efosa’s Introduction

Engineer | Programmer | Innovator

Technical Authority

Shell Nigeria

Subject Mater Expert (EMEA) for Process Automation & Control (PACO)-Subsea control systems and Subsea Distribution

Containerization 85 %

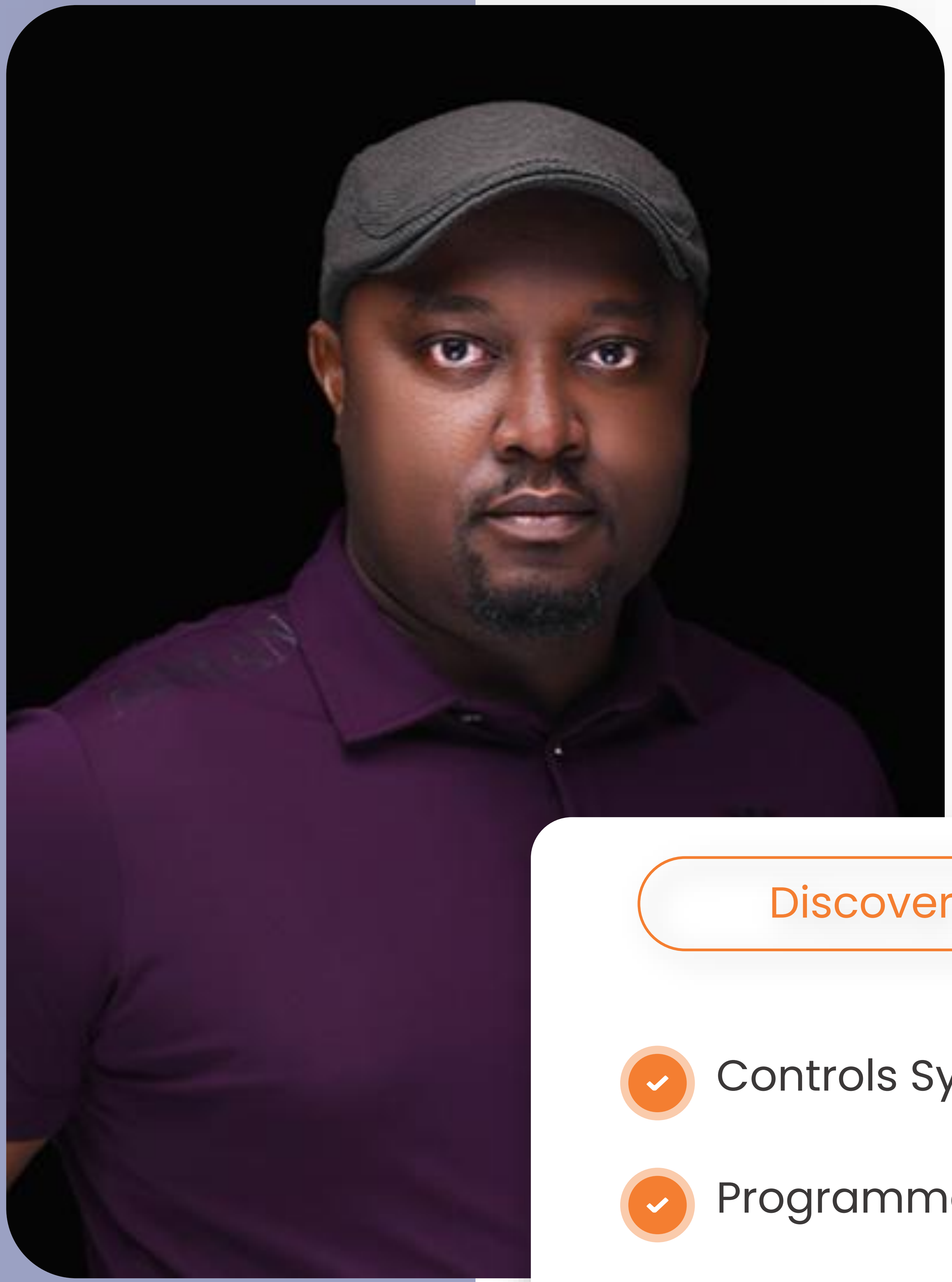
AI & ML 80 %

Renewable Energy 90 %

Innovator, VC

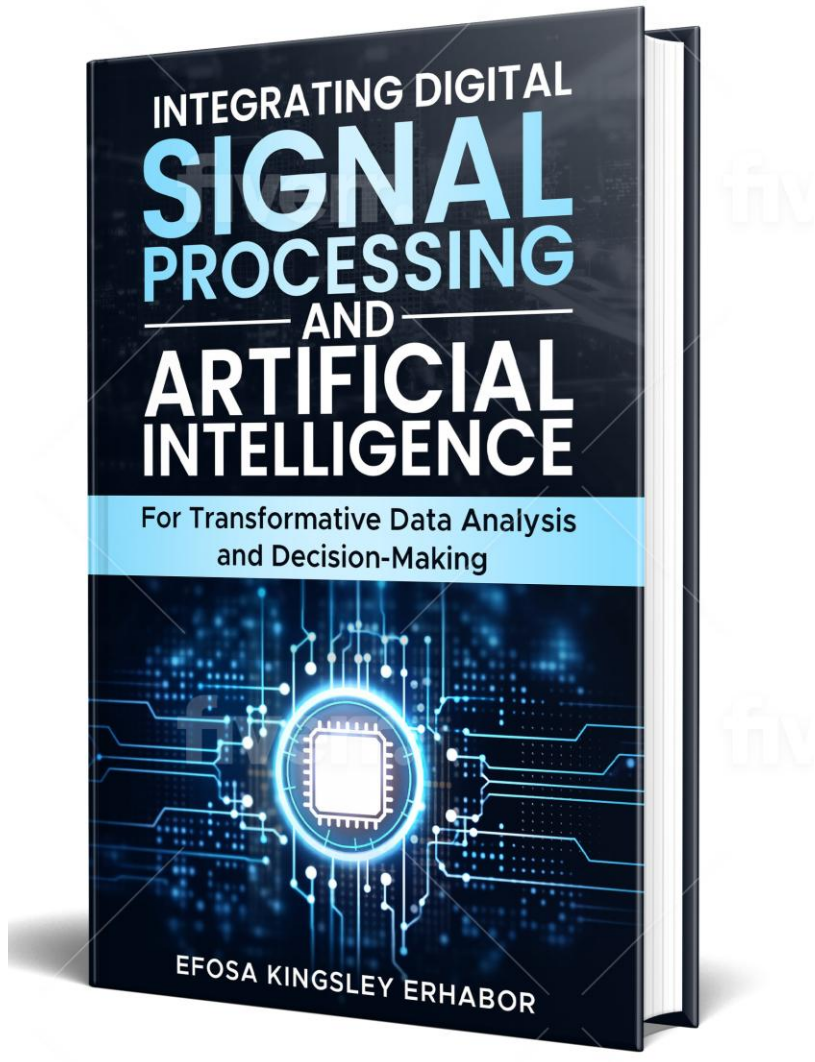
Katharos Technologies

Linux, Devops, AI and Software SME, Innovator and enterprenur



Discover

- ✓ Controls System
- ✓ Programmer
- ✓ Subsea Engineer
- ✓ AI and ML



INTEGRATING DIGITAL SIGNAL PROCESSING AND ARTIFICIAL INTELLIGENCE

For Transformative Data Analysis and Decision-Making

EFOSA KINGSLEY ERHABOR

X or Twitter

Linkedin

Learning Objectives

ELE 503: Advanced Computer Programming and Statistics

- Grasp Python syntax and fundamental programming constructs.
- Utilize libraries like NumPy and SciPy for numerical computations.
- Write Python scripts to automate engineering tasks and data analysis.

Part 1:

Python syntax and fundamental programming

Why Python for Engineers?

ELE 503: Advanced Computer Programming and Statistics

- **Advantages of Python:**
  - High readability and ease of learning.
  - Extensive libraries for scientific and numerical computing.
  - Strong community support and continuous development.
  - Integration capabilities with other languages (e.g., C#) and tools.
  - Versatility across various engineering disciplines.

Notes:

- Discuss the advantages of using Python in engineering contexts.
- Mention interoperability with languages like C#
performance-critical tasks.

•

Python vs. C# Overview

ELE 503: Advanced Computer Programming and Statistics

- **Python:**
- **Type:** Interpreted language
- **Typing:** Dynamic typing
- **Strengths:** Rapid development, extensive scientific libraries, ease of integration

- **C#:**
- **Type:** Compiled language
- **Typing:** Static typing
- **Strengths:** Performance, robust .NET ecosystem, strong type safety

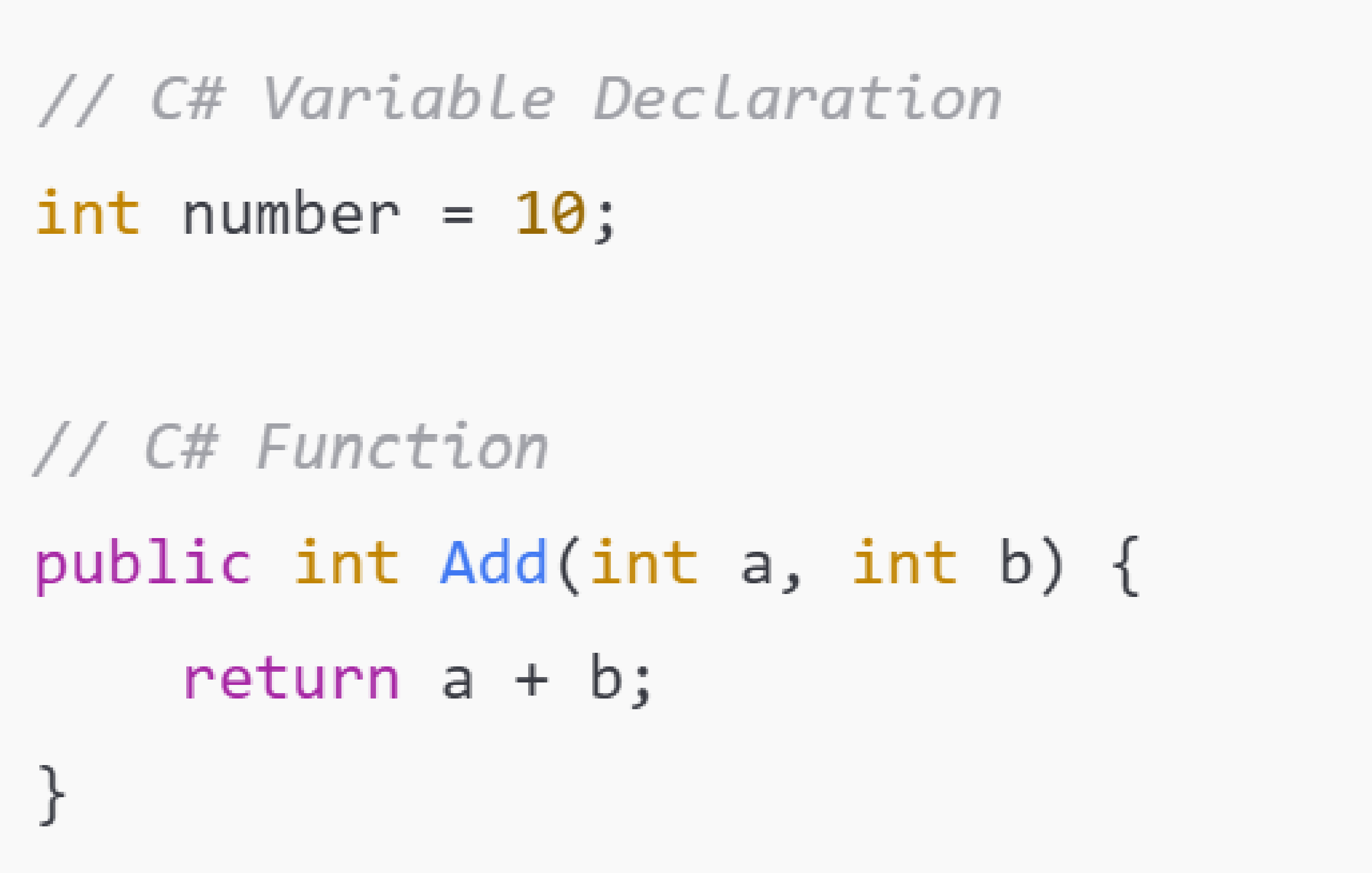
Python Syntax Basics

ELE 503: Advanced Computer Programming and Statistics

- Variables and Data Types
- Control Structures (if, for, while)
- Functions and Modules

.

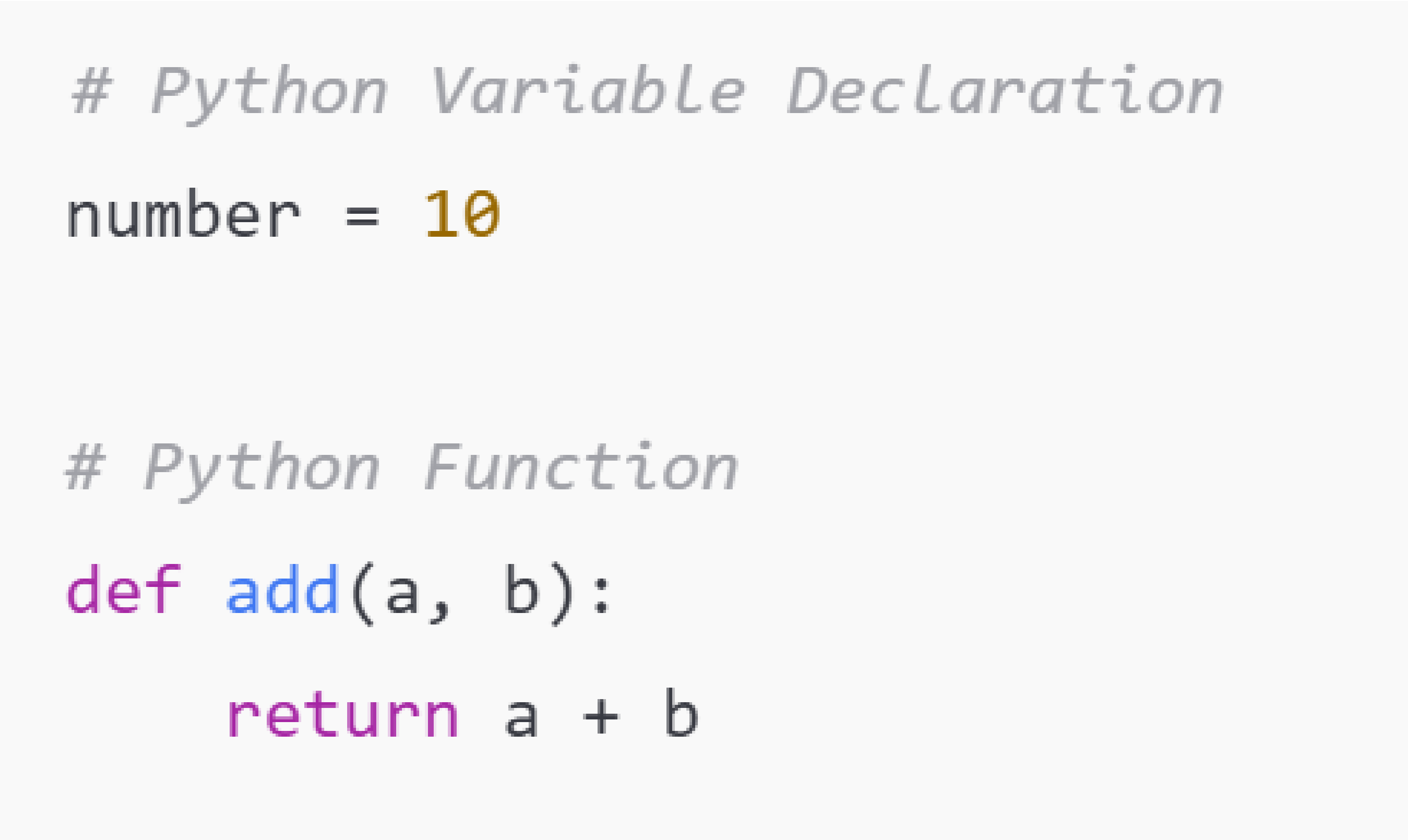
- C# Example:



```
// C# Variable Declaration
int number = 10;

// C# Function
public int Add(int a, int b) {
    return a + b;
}
```

- Python Example:



```
# Python Variable Declaration
number = 10

# Python Function
def add(a, b):
    return a + b
```

Data Structures in Python

ELE 503: Advanced Computer Programming and Statistics

- **Primary Data Structures:**
  - **Lists:** Ordered, mutable collections.
  - **Tuples:** Ordered, immutable collections.
  - **Dictionaries:** Key-value pairs, unordered.
  - **Sets:** Unordered collections of unique elements.
- **Advanced Concepts:**
  - List Comprehensions
  - Nested Data Structures

Python Control Structures

ELE 503: Advanced Computer Programming and Statistics

- **Conditional Statements:**

  if, elif, else

- **Loops:**

  for loops

  while loops

- **Exception Handling:**

  try, except, finally

Functions and Modules

ELE 503: Advanced Computer Programming and Statistics

- **Defining Functions:**
  - Syntax and usage
  - Parameters and return values
- **Importing Modules:**
  - Standard libraries
  - Third-party libraries
- **Scope and Lifetime:**
  - Local vs. Global variables

Part 2:

Python libraries

Introduction to NumPy

ELE 503: Advanced Computer Programming and Statistics

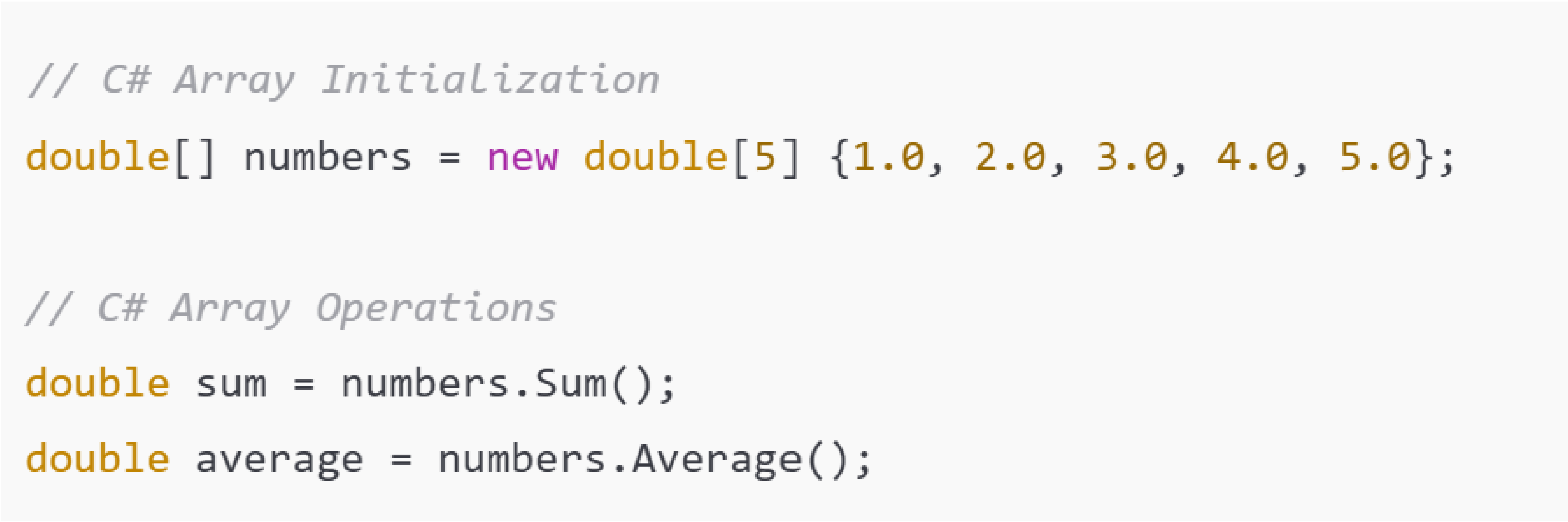
- **What is NumPy?**
  - Fundamental package for numerical computing in Python.
- **Key Features:**
  - N-dimensional arrays (Arrays vs. Lists)
  - Mathematical functions
  - Linear algebra, Fourier transforms, and random number generation

NumPy Arrays

ELE 503: Advanced Computer Programming and Statistics

- **Creating Arrays:**
  - From lists
  - Using built-in functions (arange, linspace, zeros, ones)
- **Array Indexing and Slicing:**
  - Accessing elements
  - Slicing multi-dimensional arrays
- **Array Mathematics:**
  - Element-wise operations
  - Matrix operations

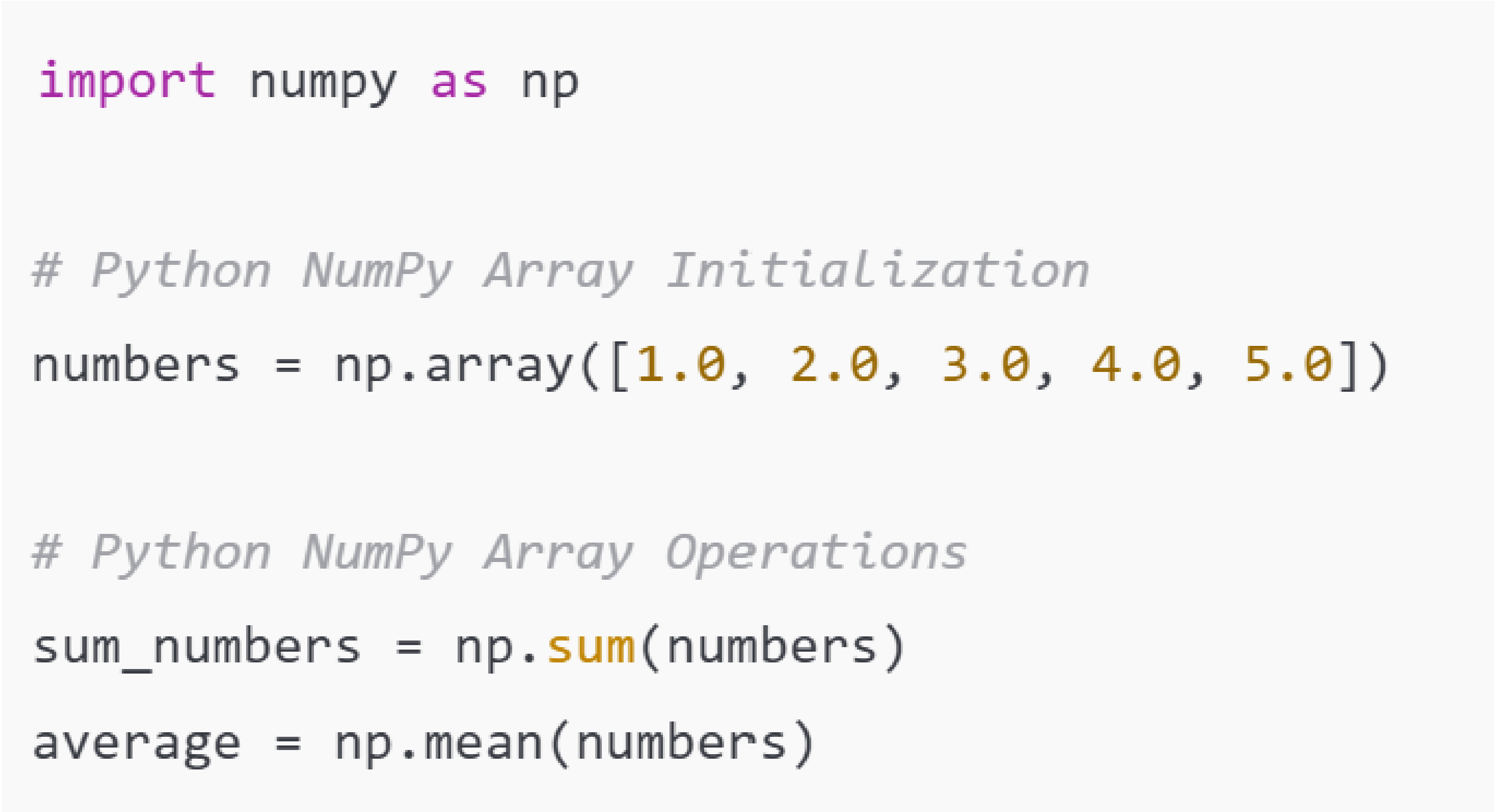
C# Example:



```
// C# Array Initialization
double[] numbers = new double[5] {1.0, 2.0, 3.0, 4.0, 5.0};

// C# Array Operations
double sum = numbers.Sum();
double average = numbers.Average();
```

Python Example:



```
import numpy as np

# Python NumPy Array Initialization
numbers = np.array([1.0, 2.0, 3.0, 4.0, 5.0])

# Python NumPy Array Operations
sum_numbers = np.sum(numbers)
average = np.mean(numbers)
```

Introduction to SciPy

ELE 503: Advanced Computer Programming and Statistics

- **What is SciPy?**
  - Open-source library for scientific and technical computing.
- **Key Modules:**
  - optimize for optimization tasks
  - integrate for integration and differential equations
  - signal for signal processing
  - stats for statistical functions

- SciPy vs. NumPy

Part 3:

Python Real World Examples

Automating Engineering Tasks with Python

ELE 503: Advanced Computer Programming and Statistics

- **Scripting Basics:**
  - Writing and executing Python scripts
  - Command-line arguments
- **File I/O Operations:**
  - Reading from and writing to files
  - Handling different file formats (CSV, JSON, XML)
- **Automating Data Processing:**
  - Batch processing
  - Scheduling scripts

Data Analysis with Python

ELE 503: Advanced Computer Programming and Statistics

- **Using Pandas for Data Manipulation:**
  - DataFrames and Series
  - Data cleaning and transformation
- **Data Visualization with Matplotlib and Seaborn:**
  - Creating plots and charts
  - Customizing visualizations
- **Statistical Analysis:**
  - Descriptive statistics
  - Inferential statistics

Solving Differential Equations

ELE 503: Advanced Computer Programming and Statistics

- **Numerical Methods in Python:**
  - Euler’s method
  - Runge-Kutta methods
- **Using SciPy’s odeint:**
  - Solving ordinary differential equations (ODEs)
- **Example: Simple Harmonic Oscillator**

$$\frac{d^2x}{dt^2} + \omega^2 x = 0$$

Optimizing Systems with Python

ELE 503: Advanced Computer Programming and Statistics

- **Optimization Techniques:**
  - Gradient descent
  - Constrained and unconstrained optimization
- **Using SciPy's optimize Module:**
  - Minimization and maximization functions
  - Solvers for different optimization problems
- **Example: Minimizing Cost Function**

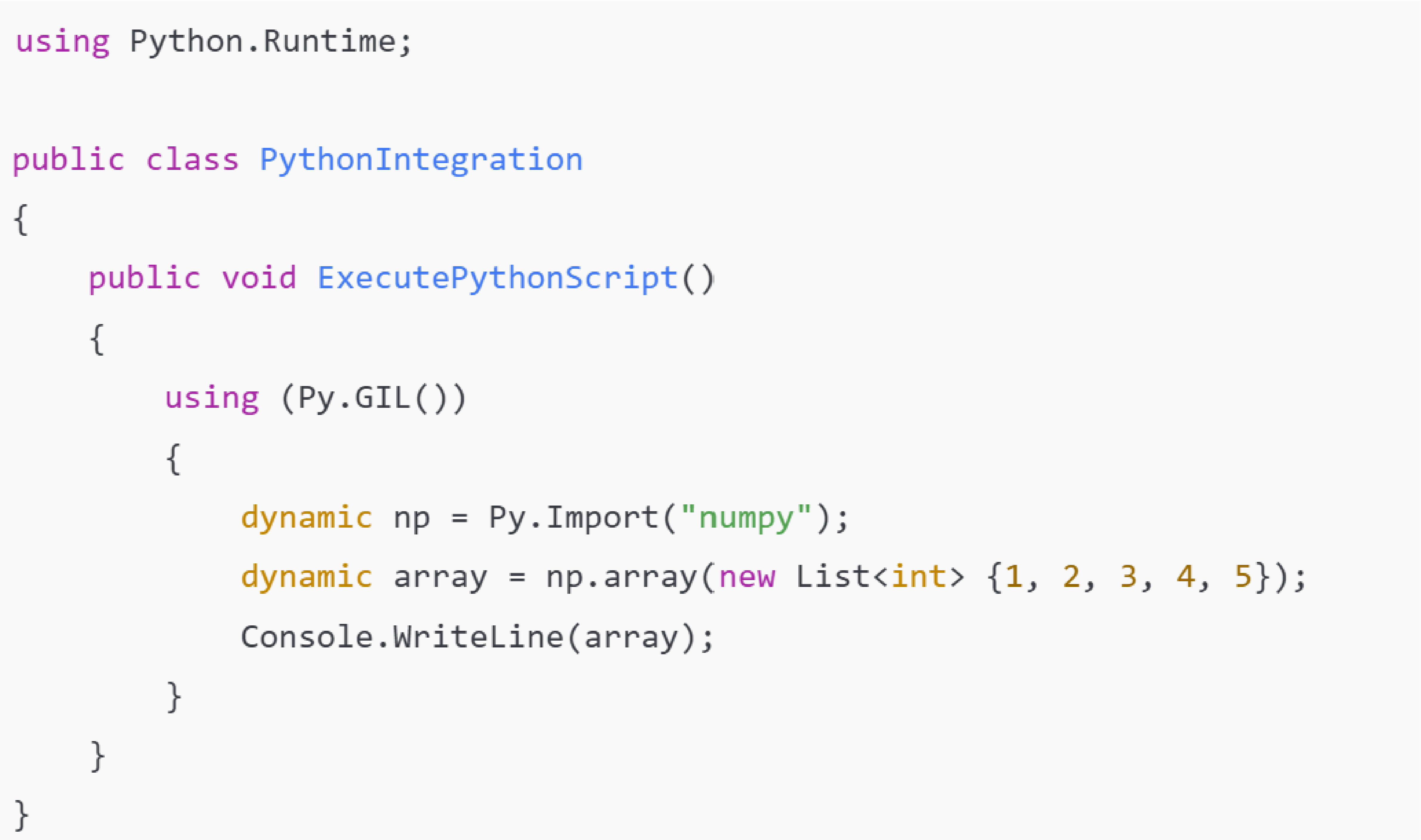
$$f(x) = x^2 + 4x + 4$$

Integration with C#

ELE 503: Advanced Computer Programming and Statistics

- **Calling Python from C#:**
  - Embedding Python scripts in C# applications
- **Using Interoperability Tools:**
  - Python.NET
  - IronPython
- **Example Use Cases:**
  - Data processing in Python with a C# frontend
  - Leveraging Python libraries within a C# project.

C# Example with Python.NET:



```csharp
using Python.Runtime;

public class PythonIntegration
{
    public void ExecutePythonScript()
    {
        using (Py.GIL())
        {
            dynamic np = Py.Import("numpy");
            dynamic array = np.array(new List<int> {1, 2, 3, 4, 5});
            Console.WriteLine(array);
        }
    }
}
```

Interactive Coding with Jupyter Notebooks

ELE 503: Advanced Computer Programming and Statistics

- **Introduction to Jupyter Notebooks:**
  - Web-based interactive computing environment
- **Benefits for Teaching and Learning:**
  - Combines code, text, and visualizations
  - Facilitates experimentation and immediate feedback
- **Basic Navigation and Features:**
  - Creating and running cells
  - Markdown support
  - Extensions and plugins

Example 1: Solving a Differential Equation

- **Title:** Solving a Simple Harmonic Oscillator with SciPy
- **Problem Statement:**
- Model the motion of a simple harmonic oscillator using the differential equation: d^2xdt^2+ω2x=0 where ω -omega is the angular frequency.

$$\frac{d^2x}{dt^2} + \omega^2 x = 0$$

- **Python Code Implementation**

Results and Interpretation:

- Output shows the minimum point of the function.
- Illustrates the use of minimize for optimization tasks.

Example 2: System Optimization

ELE 503: Advanced Computer Programming and Statistics

- **Title:** Minimizing a Cost Function with SciPy
- **Problem Statement:**
- Find the minimum of the cost function: f(x)=x^2 + 4x + 4
- **Python Code Implementation:**

Results and Interpretation:

- Output shows the minimum point of the function.
- Illustrates the use of minimize for optimization tasks.

Example 3: Data Analysis Pipeline

ELE 503: Advanced Computer Programming and Statistics

•**Title:** Analyzing Engineering Data with Pandas and Matplotlib

•**Problem Statement:**

•Analyze and visualize temperature data collected over time.

•**Results and Interpretation:**

•Visual representation of temperature trends and moving averages.

•Demonstrates data manipulation and visualization techniques.

Example 4: Automating a Task

ELE 503: Advanced Computer Programming and Statistics

- **Title:** Automating File Processing with Python
- **Problem Statement:**
- Automatically rename and organize engineering report files.

- **Results and Interpretation:**
- Files are renamed systematically.
- Saves time by automating repetitive file operations.

Example 5: Visualization of Engineering Data

ELE 503: Advanced Computer Programming and Statistics

- **Title:** Visualizing Stress-Strain Curves with Matplotlib
- **Problem Statement:**
- Plot stress-strain curves for different materials.

- **Results and Interpretation:**
- Clear comparison of stress-strain behavior between materials.
- Useful for material selection and analysis in engineering design.

Part 4:
Best Practices in Python Programming

Best Practices in Python Programming

ELE 503: Advanced Computer Programming and Statistics

- **Code Readability and PEP 8:**
  - Adhering to Python's style guidelines.
- **Writing Modular and Reusable Code:**
  - Functions, classes, and modules.
- **Documentation and Comments:**
  - Importance of clear documentation.
  - Using docstrings and inline comments.

Exploring Python's Engineering Libraries

ELE 503: Advanced Computer Programming and Statistics

- **Overview of Key Libraries:**
  - **Pandas:** Data manipulation and analysis.
  - **Matplotlib:** Data visualization.
  - **SciPy:** Scientific and technical computing.
  - **SymPy:** Symbolic mathematics.
  - **TensorFlow/PyTorch:** Machine learning and deep learning.

- **How to Choose the Right Library:**
  - Based on project requirements.
  - Community support and documentation.
  - Performance considerations.

Conclusion and Further Resources

ELE 503: Advanced Computer Programming and Statistics

- **Recap of Learning Objectives:**
  1. Grasp Python syntax and fundamental programming constructs.
  2. Utilize libraries like NumPy and SciPy for numerical computations.
  3. Write Python scripts to automate engineering tasks and perform data analysis.

- **Summary of Key Points:**
  - Python’s versatility and powerful libraries make it indispensable for engineers.
  - Understanding basic syntax is crucial for leveraging advanced features.
  - Practical examples demonstrate Python’s applicability in solving real-world engineering problems.

Conclusion and Further Resources

ELE 503: Advanced Computer Programming and Statistics

- **Recommended Resources:**
  - **Official Python Documentation:** python.org
  - **NumPy Documentation:** numpy.org
  - **SciPy Documentation:** scipy.org
  - **Pandas Documentation:** pandas.pydata.org
  - **Online Tutorials and Courses:** Coursera, edX, Udemy
  - **Community Forums:** Stack Overflow, Reddit’s r/learnpython