MentorHints: Python Advanced Syllabus

---

1. Python Basics

- **Introduction to Python**
- Python runtime environment: Interpreter and execution flow
- Installation: Python setup, virtual environments, package managers (pip)
- **Variables and Literals**
- Variable declaration, naming conventions, dynamic typing
- Literals: Integers, floats, strings, booleans
- **Basic Input and Output**
- Input handling with input()
- Output formatting with f-strings, format(), and % operator
- **Type Casting**
- Implicit and explicit type conversion

---

2. Course Tools and Environment Setup

- **Version Control with Git and GitHub**
- Repository setup and exercise management
- Collaboration workflows (fork, pull requests)
- **Essential Git Commands**
- clone, commit, push, pull, branch, merge, rebase
- **IDE Setup**
- Visual Studio Code: Installation, Python extensions, debugging configuration

---

3. Python Intermediate Fundamentals

- **Data Types**
- Lists: Slicing, comprehensions, nested lists
- Tuples: Immutability, unpacking
- Sets: Set operations, frozen sets

- Dictionaries: Nested dictionaries, defaultdict, OrderedDict

- **Control Flow**

- Conditional statements: Nested if-else, ternary operators
- Loops: Advanced for/while loops, loop optimization

- Break, continue, pass in complex logic

- **Functions**

- Variable scopes, global/nonlocal keywords
- Modules, namespaces, and package creation

- Importing strategies (relative, absolute)

- **Closures**

- Creating and applying closures in real-world scenarios

- **Decorators**

- Writing custom decorators, chaining decorators

- **Property Decorators**

- Getter, setter, deleter for attribute management

---

4. Python Advanced Concepts

- **Exception Handling**
- Built-in exceptions, custom exception classes

- Assertions for debugging and validation

- **File Handling**

- Text files: .txt, .csv, .tsv, .json (parsing and writing)

- Binary files: .xlsx (pandas), .pickle (serialization)

- **Object-Oriented Programming**

- Inheritance: Single and multiple inheritance
- Polymorphism: Method overriding, dynamic dispatch
- Operator overloading: Customizing operators

- Abstract classes: With and without properties

- **Functional Programming**

• Lambda functions, map/filter/reduce

• List comprehensions, generators, iterators

• **Command-Line Arguments**

• Argparse for CLI applications

• **Regular Expressions**

• Pattern matching, validation, and text processing

• **Namespaces and Package Management**

• Scopes, module resolution

• PIP, virtual environments, dependency management

• **Design Patterns and Refactoring**

• Singleton, Factory, Observer patterns

• Refactoring custom packages for reusability

• **Asynchronous Programming**

• Asyncio basics, coroutines, event loops
• Concurrent task management

---

5. Python Applications - Data Analysis and Visualization (EDA)

• **NumPy**
• Multidimensional arrays, indexing/slicing

• Broadcasting, file I/O (.npy, .npz)

• **Pandas**

• Series and DataFrames: Creation, manipulation
• Filtering, sorting, GroupBy, aggregation

• Merging/joining datasets

• **Visualization**

• Matplotlib.Pyplot: Bar charts, histograms, scatter plots
• Seaborn: Box plots, heatmaps, advanced styling
• Custom plotting techniques for publication-ready visuals

---

6. Python Applications - Artificial Intelligence

- **Machine Learning with Scikit-Learn**
- Supervised learning: Regression, classification
- Unsupervised learning: Clustering, dimensionality reduction
- Reinforcement learning: Basic concepts, Q-learning, RL algorithms
- Model training, evaluation, pipelines, hyperparameter tuning

- **Neural Networks**

- Architectures: Perceptrons, CNNs, RNNs, LSTM
- Activation functions, loss functions, optimizers

- **PyTorch**

- Tensors, autograd, dynamic computation graphs
- Building/training neural networks, custom datasets
- GPU acceleration, model optimization

- **TensorFlow**

- Keras API: Model building, training, deployment
- Custom layers, callbacks, TensorBoard visualization

- **Web APIs for AI Models**

- FastAPI/Flask: Model serving, RESTful endpoints
- Integration with ML frameworks, request handling

- **Transfer Learning with Hugging Face**

- Pre-trained models: Transformers for NLP
- Fine-tuning concepts: Objective, process, and applications

- **Natural Language Processing**

- Text preprocessing: Tokenization, stemming, lemmatization

- **AI/ML Project**

- End-to-end AI/ML project with FastAPI