**Week 1: Python Basics**

**Day 1: Introduction to Python**

* Install Python and set up the development environment
* Python syntax, variables, and data types
* Basic input/output operations

**Day 2: Control Structures**

* Conditional statements (if, elif, else)
* Loops (for, while)
* Break and continue statements

**Day 3: Functions**

* Defining and calling functions
* Function arguments and return values
* Lambda functions

**Day 4: Data Structures - Lists and Tuples**

* List operations and methods
* Tuple operations
* List comprehensions

**Day 5: Data Structures - Sets and Dictionaries**

* Set operations and methods
* Dictionary operations and methods
* Dictionary comprehensions

**Day 6: Strings and String Manipulation**

* String operations and methods
* String formatting
* Regular expressions

**Day 7: Review and Practice**

* Solve coding exercises on platforms like LeetCode or HackerRank
* Review the concepts learned in Week 1

**Week 2: Advanced Python Concepts**

**Day 8: File Handling**

* Reading and writing files
* Working with file paths
* Exception handling

**Day 9: Modules and Packages**

* Importing modules and using standard libraries
* Creating and using custom modules
* Package installation with pip

**Day 10: Object-Oriented Programming (OOP) - Part 1**

* Classes and objects
* Instance variables and methods
* Class variables and methods

**Day 11: OOP - Part 2**

* Inheritance and polymorphism
* Method overriding
* Encapsulation and abstraction

**Day 12: Iterators and Generators**

* Understanding iterators and the iter() function
* Creating and using generators
* Generator expressions

**Day 13: Decorators and Context Managers**

* Function decorators
* Class decorators
* Using with statement and context managers

**Day 14: Review and Practice**

* Work on small projects or coding challenges
* Review the concepts learned in Week 2

**Week 3: Libraries and Frameworks**

**Day 15: NumPy and Pandas**

* Introduction to NumPy: arrays, operations
* Introduction to Pandas: DataFrames, operations
* Basic data manipulation and analysis

**Day 16: Matplotlib and Seaborn**

* Plotting with Matplotlib
* Data visualization with Seaborn
* Creating different types of plots and charts

**Day 17: Web Scraping with BeautifulSoup and Requests**

* Basics of web scraping
* Using requests to fetch web pages
* Parsing HTML with BeautifulSoup

**Day 18: Introduction to Flask (Web Framework)**

* Setting up a Flask project
* Creating routes and templates
* Handling forms and user input

**Day 19: APIs and RESTful Services**

* Understanding APIs
* Consuming APIs with requests
* Creating a simple RESTful API with Flask

**Day 20: Review and Mini-Project**

* Work on a mini-project using the libraries/frameworks learned
* Review all concepts learned
* Prepare for future learning paths