Python Beginner Projects

Project 1: Weather Data Analyzer

Objective: Develop a Python project using functions, NumPy, Pandas, and API requests to fetch and analyze live weather data for multiple cities.

1. **Step 1: Environment Setup** - Install Python and required libraries: requests, pandas, numpy. Create a new Python file and obtain an API key from OpenWeatherMap.
2. **Step 2: Define Functions** - Create functions to call the API, extract temperature, humidity, and weather descriptions, and store structured results.
3. **Step 3: Make API Requests** - Fetch data for multiple cities, handle missing or invalid responses, and store results in a list.
4. **Step 4: Data Handling Using Pandas** - Convert data into a Pandas DataFrame, inspect columns, and save the file as CSV.
5. **Step 5: Numerical Analysis Using NumPy** - Use NumPy to calculate average, max, and min temperatures, and identify patterns or trends.
6. **Step 6: Generate Output and Insights** - Display a summary table with insights and optional visualizations using Matplotlib.
7. **Step 7: Documentation and Submission** - Include objectives, tools, outcomes, screenshots, and well-commented code.

Project 2: Student Grade Calculator

Objective: Create a Python program to calculate total, average, and grade for multiple students using core Python features.

1. **Step 1: Define the Objective** - Build a system to calculate grades for multiple students using inputs, loops, conditionals, and functions.
2. **Step 2: Environment Setup** - Open a Python file or Jupyter Notebook; no external libraries required.
3. **Step 3: Collect Input** - Accept student names and subject marks, and store them in a list or dictionary.
4. **Step 4: Use Functions** - Write functions to calculate total, average, and grade (A, B, C, D, Fail).
5. **Step 5: Apply Control Statements** - Use if-else for grade classification and loops for multiple students.
6. **Step 6: Display Output** - Print a formatted summary showing name, total, average, and grade, and optionally show topper and class average.
7. **Step 7: Validate and Test** - Test with at least three students and handle invalid inputs.

8 **Step 8: Documentation** - Explain code parts briefly and mention Python concepts used (functions, conditionals, data structures).