Course Title: Advanced Geospatial Programming with Python

- **Duration:** 5 Days (approximately 35-40 instructional hours)
- **Course Goal:** To provide experienced GIS professionals and Python programmers with a deep, practical mastery of the open-source geospatial Python ecosystem. This course will move beyond basic scripting to cover advanced data manipulation, analysis, and visualization of both vector and raster data using core libraries like GeoPandas, Rasterio, Shapely, and Fiona, enabling participants to build powerful, scalable, and automated geospatial workflows.
- **Core Focus:** Mastering the "py-geo" stack to perform complex GIS tasks programmatically, replacing manual, GUI-based workflows with efficient and repeatable code.

Learning Objectives:

Upon successful completion of this workshop, participants will be able to:

- **Master** the GeoPandas library for sophisticated vector data manipulation, analysis, and I/O.
- **Perform** complex geometric operations and spatial queries using Shapely and GeoPandas.
- **Read, write, and manipulate** raster datasets programmatically using Rasterio.
- **Conduct** raster algebra and zonal statistics by integrating Rasterio with NumPy.
- **Handle** and transform between different coordinate reference systems (CRS) effectively using PyProj.
- **Create** publication-quality static and interactive maps directly from Python.
- **Build** custom, multi-step geospatial workflows that integrate multiple data sources and libraries.
- **Write** clean, efficient, and well-documented Python code for geospatial tasks.
- **Develop** a capstone project that solves a complex spatial problem from start to finish using a scripted workflow.

Target Audience:

- GIS Analysts and Specialists who want to automate their work and perform more advanced analysis.
- Data Scientists and Data Analysts who need to incorporate geospatial data into their workflows.
- Remote Sensing Scientists and Environmental Modelers.
- Geographers and researchers who use quantitative spatial methods.
- Anyone with a solid Python and GIS foundation looking to become a power user of the open-source geospatial stack.

Prerequisites:

- **Strong intermediate-level Python programming skills are absolutely essential.** Participants MUST be comfortable with functions, loops, conditional statements, and data structures (lists, dictionaries).
- **Solid experience with the Pandas library is required.** This course builds heavily on Pandas concepts.
- **A strong foundational understanding of GIS concepts is required.** Participants must understand vector vs. raster data, coordinate systems, projections, and common geoprocessing operations (e.g., buffer, intersect, clip).

Teaching Methodology:

This is an intensely hands-on, "code-along" workshop in a Jupyter Notebook environment. The methodology includes:

- **Interactive Lectures & Library Deconstruction:** Introducing each library's architecture and key functionalities.

ADEPT TRAINING CENTER

- **Live Coding Demonstrations:** The instructor will write and explain complex geospatial code live.
- **"Coding Labs":** The core of the course. Participants will spend the majority of their time writing Python code to solve a series of progressively difficult geospatial problems.
- **Problem-Based Learning:** Labs are framed as real-world analytical tasks.
- **Code Reviews & Best Practices:** Discussing different ways to solve a problem and focusing on writing clean, efficient, "Pythonic" code.
- **Capstone Project:** An integrated project that requires participants to combine skills and libraries from across the week.

*(For **Online Delivery**, this is highly effective. It requires a shared coding environment (like Google Colab or a pre-configured Binder instance) and extensive use of screen sharing, breakout rooms for coding challenges, and a chat channel for debugging.)*

Materials Provided:

- Comprehensive digital workbook with key concepts, library documentation summaries, and detailed code examples.
- A complete set of **Jupyter Notebooks** with all lab exercises and fully commented solutions.
- All necessary geospatial datasets (shapefiles, GeoJSON, GeoTIFFs) for the labs and capstone project.
- A "Geospatial Python Cheat Sheet" for key libraries.
- Certificate of Completion.

Detailed Day-by-Day Curriculum

Day 1: The Vector Data Powerhouse – GeoPandas

- **Module 1: The Geospatial Python Ecosystem & Course Setup (Approx. 1.5 hours)**
  - Welcome, introductions, and course objectives.
  - Overview of the open-source "py-geo" stack (GeoPandas, Rasterio, Shapely, Fiona, PyProj, etc.) and how they fit together.
  - Setting up the Conda environment and installing key libraries.
  - Recap of key Pandas concepts.
- **Module 2: Introduction to GeoPandas (Approx. 3 hours)**
  - The GeoDataFrame: The core data structure.
  - The "geometry" column and Shapely geometry objects (Point, LineString, Polygon).
  - Reading and writing geospatial vector data (Shapefiles, GeoJSON, GeoPackage).
  - Basic plotting and visualization with .plot().
- **Module 3: Coordinate Reference Systems (CRS) in GeoPandas (Approx. 3 hours)**
  - The critical importance of CRS.
  - Understanding CRS information in a GeoDataFrame.
  - **Reprojecting** data from one CRS to another (.to_crs()).
  - The role of PyProj and understanding EPSG codes.
  - **Hands-on Lab:** Loading multiple datasets, checking their CRS, and reprojecting them to a common system for analysis.
  - **End of Day 1 Recap & Q&A.**

Day 2: Advanced Vector Analysis & Spatial Joins

ADEPT TRAINING CENTER

- **Module 4: Geometric Operations & Attribute-Based Queries (Approx. 2.5 hours)**
    - Accessing geometric properties (.area, .length, .centroid).
    - Creating new geometries: Buffering (.buffer()), convex hulls, simplifying geometries.
    - Combining Pandas-style attribute queries with spatial queries.
- **Module 5: Spatial Joins with GeoPandas (Approx. 3 hours)**
    - The power of the spatial join.
    - **Spatial Relationships:** Intersects, Contains, Within, Touches.
    - Performing spatial joins to transfer attributes based on location (e.g., assigning census data to school districts).
    - Understanding left vs. inner spatial joins.
    - Optimizing spatial joins using spatial indexes.
    - **Intensive Hands-on Lab:** A multi-step lab involving buffering, attribute queries, and spatial joins to answer a complex question (e.g., "Find the total population living within 1km of a polluted river").
- **Module 6: Overlay Analysis & Set-Theoretic Operations (Approx. 2 hours)**
    - Performing vector overlay analysis: overlay() function.
    - **Intersection, Union, Identity, Symmetrical Difference.**
    - Using overlay to combine geometric features and their attributes.
    - **End of Day 2 Recap & Q&A.**

Day 3: Mastering Raster Data with Rasterio

- **Module 7: Introduction to Rasterio for Raster I/O (Approx. 2.5 hours)**
    - The raster data model revisited (bands, pixels, geotransform, CRS).
    - Opening, reading metadata from, and closing raster files (e.g., GeoTIFFs) with Rasterio.
    - Reading raster bands into NumPy arrays.
    - Writing NumPy arrays back to a new GeoTIFF file, preserving georeferencing.
- **Module 8: Raster Algebra & The Power of NumPy (Approx. 3 hours)**
    - The core of raster analysis: Performing mathematical operations on raster arrays.
    - **Map Algebra:** Applying arithmetic, logical, and conditional operations on a cell-by-cell basis.
    - Calculating spectral indices like **NDVI** programmatically.
    - Reclassifying raster data using NumPy functions.
    - **Intensive Hands-on Lab:** Participants will read multiple bands of a satellite image, calculate NDVI, and reclassify the result into low, medium, and high vegetation classes.
- **Module 9: Working with Raster Masks & Vector Data (Approx. 2 hours)**
    - The rasterio.mask module.
    - **Clipping (Masking):** Using a vector polygon (e.g., a park boundary) to clip a raster dataset to a specific area of interest.
    - The process of converting vector features into a raster mask.
    - **Hands-on Lab:** Clipping a DEM raster to a watershed boundary.
    - **End of Day 3 Recap & Q&A.**

Day 4: Advanced Raster Analysis & Visualization

ADEPT TRAINING CENTER

- **Module 10: Zonal Statistics (Approx. 3 hours)**
    - The concept of zonal statistics: Summarizing raster values within zones defined by a vector layer.
    - Using the rasterstats library to easily perform zonal statistics.
    - Calculating statistics like mean, median, sum, and majority for each zone (e.g., "Calculate the average elevation for each land cover type").
    - Joining the statistical results back to the vector data.
    - **Hands-on Lab:** Calculating the average NDVI for a set of farm field polygons.
- **Module 11: Advanced Visualization: Static & Interactive Maps (Approx. 3 hours)**
    - Creating publication-quality static maps with **Matplotlib** and GeoPandas.
    - Customizing map elements: Titles, legends, scale bars, north arrows.
    - Layering multiple vector and raster datasets.
    - Introduction to creating interactive web maps with libraries like **Folium** or **hvPlot**.
    - **Workshop:** Participants create a high-quality static map of their analytical results and then create a simple interactive web map.
- **Module 12: Performance & Scalability (Approx. 1.5 hours)**
    - Strategies for working with large geospatial datasets that don't fit in memory.
    - The concept of chunking and parallel processing with Dask (overview).
    - The role of efficient file formats like GeoPackage and Cloud-Optimized GeoTIFFs (COGs).
    - **End of Day 4 Recap & Q&A.**

Day 5: Building Workflows & Capstone Project

- **Module 13: Building an End-to-End Geospatial Workflow (Approx. 2.5 hours)**
    - The importance of structuring a geospatial project with Python.
    - Writing modular, reusable functions for common GIS tasks.
    - Combining vector and raster processing in a single script.
    - Best practices for project organization and documentation.
- **Module 14: Capstone Project (Approx. 4 hours)**
    - **Final Project:** Participants are given a new, complex, multi-stage problem and a set of raw geospatial data.
    - The problem will require them to integrate skills from the entire week. Example: "Given a DEM, land cover data, and locations of towns, identify potential sites for a new wildfire monitoring tower that are on high ground, in a non-forest area, and within 5km of a town."
    - Participants must write a clean, well-documented Jupyter Notebook that performs the entire analysis and produces a final map of suitable locations.
- **Module 15: Capstone Presentations & Code Review (Approx. 1 hour)**
    - Participants briefly present their capstone project approach and results.
    - Group discussion on different approaches and code optimization.
- **Module 16: The Future of Geospatial Python & Course Wrap-up (Approx. 0.5 hours)**
    - Brief overview of other important libraries and future trends (e.g., spatial databases, cloud-native geospatial, machine learning with geospatial data).
    - Open Q&A, course evaluation, and closing.

ADEPT TRAINING CENTER