**Step 01: Import and Authenticate Earth Engine**

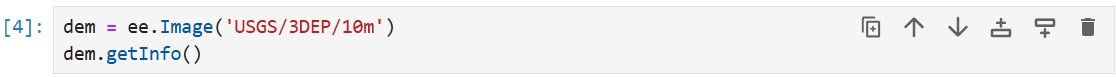
This section imports the Google Earth Engine (GEE) Python API, authenticates the user to access Earth Engine services, and initializes the API with a specific GEE project ID (geog4057spring2025). Authentication ensures the user has access rights to interact with Earth Engine datasets and services.

A screenshot of a phone

AI-generated content may be incorrect.

**Step 02: Load Digital Elevation Model (DEM)**

This loads the USGS 3DEP 10-meter resolution DEM as an Earth Engine image object, which contains elevation data across the U.S. The getInfo() call retrieves metadata about the image, such as projection, bands, and tile structure.



**Step 03: Extract Elevation at a Specific Point**

A white background with numbers and letters

AI-generated content may be incorrect.

This defines a **point geometry** using longitude and latitude coordinates and wraps it into an **Earth Engine FeatureCollection,** which is required for spatial sampling.

A screen shot of a computer code

AI-generated content may be incorrect.

This samples the DEM elevation value at the specified point using the sampleRegions() method and retrieves the output using getInfo(). The resulting dictionary includes the elevation value at that location (e.g., 3.22 meters above sea level).

**Step 4: Read Coordinates from a CSV File**

This code block reads the CSV file containing point location data using pandas. The dataset includes columns for column and row indices along with X (longitude) and Y (latitude) coordinates. This table will be used to extract elevation values at each point.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 5: Check Spatial Reference from Raster**

A close-up of a sign

AI-generated content may be incorrect.

In this step, the code retrieves the spatial reference system (CRS) of the flood classification raster using ArcPy. The factoryCode 32119 indicates that the raster uses NAD\_1983\_StatePlane\_North\_Carolina\_FIPS\_3200, which ensures consistent alignment with the elevation points later extracted.

**Step 6: Convert Table to Point Shapefile and Add Elevation Field**

This step converts the input CSV table into a GeoDataFrame using X and Y coordinates, assigns it to the same coordinate system as the raster (EPSG:32119), and exports it as a shapefile. An empty elevation field of type float is then added using ArcPy to store values extracted from GEE.

A screenshot of a computer program

AI-generated content may be incorrect.

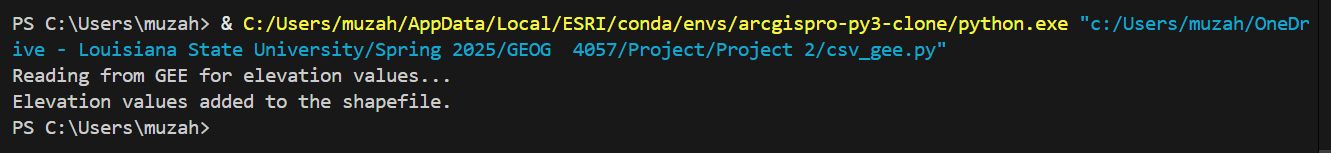
**Step 7: Extract and Write Elevation to Shapefile from Google Earth Engine**

A screen shot of a computer code

AI-generated content may be incorrect.

This block loops through each point in the shapefile using arcpy.da.UpdateCursor, creates a geometry object from its coordinates, and sends it to Google Earth Engine (ee.Geometry.Point) to sample elevation from the DEM (dem.sampleRegions). The returned elevation value is parsed from the response and written back into the elevation field of the shapefile.

**Step 8: Running the csv\_gee.py Script to Add Elevation to the Shapefile**



The csv\_gee.py script was executed using the arcgispro-py3-clone environment in VS Code. It authenticated Earth Engine, accessed the USGS/3DEP/10m DEM, read point coordinates from a CSV, and created a shapefile with the specified coordinate system (EPSG:32119). Elevation values were sampled from Earth Engine and written to the shapefile. The process completed successfully, as confirmed by terminal output.

**Step 9: Developing and Executing a Python Toolbox in ArcGIS Pro**

In the final step, a custom Python toolbox named project2\_test.pyt was developed and integrated within ArcGIS Pro. This toolbox script was modified and designed according to the class lecture to automate the elevation extraction process by connecting a CSV file containing X, Y coordinates with Google Earth Engine's 10-meter resolution DEM (USGS/3DEP/10m). Upon execution, the toolbox read the input CSV, sampled elevation values from Earth Engine, and generated a shapefile (boundary.shp) enriched with the elevation attribute. The output was visually verified on the ArcGIS map canvas, confirming the successful extraction and mapping of elevation data.

A screenshot of a computer

AI-generated content may be incorrect.