# GISC 2335 Class Notes - March 4, 2025

Instructor: Mr. Erik Bushland Week: 7 Topic: Chapter 6 – Exploring Spatial Data

## Housekeeping & Announcements

* Today is Week 7. Next week is Spring Break.
* Midterm Review will be on Tuesday after Spring Break.
* Midterm Exam:

- Opens: Wednesday, March 19, at 7:00 AM

- Closes: Sunday, March 23, at 11:59 PM

- Open book, open notes.

- 3 hours to complete.

* **📌 Grading Reminder**
* As of **4:45 PM on March 4**, Mr. Bushland said he is **caught up on grading**.
* If you submitted something and **don’t see a grade or feedback**, check:
  + If he left a **comment** and follow up.
  + If there’s **no comment and no grade**, email or message him ASAP to resolve it.

## Assignments

* Chapter 5 Exercise - Due March 4 by 5 PM.
* Chapter 6 Exercise - Due Tuesday, March 18 by 5 PM.
* Submit via correct Week 6 link in course portal.
* **📌 1. Lab Materials Tip**
* Be sure to use the **correct link** for lab exercises. The **correct version should show a 2024 date**.
* The book may have other lab files, but some are outdated due to ArcGIS Pro software updates.
* **📌 2. Submitting Exercises**
* When submitting Chapter 6 exercise:
  + If it asks to submit a **project file**, you’ll need to upload your actual GIS project to the organization (likely via ArcGIS Online or a shared network).
  + If it only asks for a **Word document + Python script (.py)**, you can just submit those.
  + **Note:** Mr. Bushland prefers having the Python script too, because he may run it for verification (especially if your screen captures are unclear), but it’s not required if your screenshots are thorough.

## Key Concepts from the Lecture

### Checking for Existence of Data

* + Use arcpy.Exists() to validate before running scripts. Prevents errors due to missing datasets.

### System Paths vs. Catalog Paths

* + System paths are recognized by Windows, while Catalog paths are internal to ArcGIS Pro.

### Describing Datasets

* + Use arcpy.Describe() or arcpy.da.Describe() to get dynamic properties of GIS data.

### Versioning & Collaboration

* + Essential in environments with multiple users. Keeps shared data consistent.

### Working with Lists (Batch Processing)

* + Use arcpy.ListFeatureClasses() with wildcards and for loops for efficient data processing.

### List Comprehensions

* + A concise way to build lists using one-liners. Always returns a list.

### Workspaces Matter

* + Set arcpy.env.workspace before listing or processing spatial data.

### Comparing System and Catalog Paths

* + Use .shp = system path; .gdb = catalog path.

### Counting Datasets

* + Use len(arcpy.ListFeatureClasses()) to count features.

## Key Takeaways

|  |  |
| --- | --- |
| Concept | Key Tool / Command |
| Check if dataset exists | arcpy.Exists() |
| Describe dataset | arcpy.Describe() or arcpy.da.Describe() |
| List datasets | arcpy.ListFeatureClasses() |
| Set workspace | arcpy.env.workspace |
| Wildcards | \* |
| Batch processing | for loops |
| List comprehension | [expression for item in iterable if condition] |
| Count items in list | len() |

## Suggested Questions (Career/Academic Interest Tie-in)

### Public Health + GIS

• How could spatial data validation improve public health projects, such as mapping disease outbreaks or environmental hazards?

• Could versioning (like in ArcGIS) be adapted to health data management to track changes over time?

### Data Science + Python

• Would you recommend using list comprehensions or traditional loops for processing large spatial datasets?

• Could arcpy.Describe() be useful for automated metadata generation in data pipelines?

### AI + GIS

• How does pre-processing spatial data (checking existence, describing, listing files) help in training AI models?

• Are there Python libraries like Geopandas better suited for AI pipelines than ArcPy?

**📌 Comments in Code**

* It's **good practice** to break your code into small, readable chunks using **comments**.
* Example:

python

CopyEdit

# This section sets the workspace

arcpy.env.workspace = "C:/Data"

# This section lists feature classes

fcs = arcpy.ListFeatureClasses()

**📌 Code Clarity vs. Brevity**

* Sometimes it’s better to write longer, more detailed code to understand what's happening.
* Other times, shorter code (like **list comprehensions**) is more efficient.
* Just like math: sometimes you show every step; other times, you skip ahead.

**📌 Directory and Wildcard Example**

* When using dir in command prompt:
  + dir D\* → Lists folders that start with D
  + \*.shp → Lists all shapefiles
  + Dallas.\* → Could return any file that starts with “Dallas”

**📌 Error Prevention**

* ArcGIS tool dialog boxes automatically prevent invalid inputs (e.g., wrong data type).
* But in Python scripts, **you must manually validate** using arcpy.Describe() or arcpy.Exists().