Spatial Programming Lab 6

Seth Opatz

(Python scripts for all 8 questions attached with assignment submission)

Question 1 script:

```
# Lab 6 Question 1
import arcpy
import os
arcpy.env.overwriteOutput = True
wkspace = 'C:/PythonPro/Exercise06'
data_folder = 'C:/PythonPro/Exercise06/Lab06Data'
new_gdb = 'BlackHills.gdb'
arcpy.CreateFileGDB_management(wkspace, new_gdb)
arcpy.env.workspace = data_folder
ftr_classes = arcpy.ListFeatureClasses()
for fc in ftr_classes:
    fc_name = arcpy.da.Describe(fc)["baseName"]
    new_fc = os.path.join(wkspace, new_gdb, fc_name) #copy paths to new gdb
    arcpy.CopyFeatures_management(fc, new_fc) #copy features to new gdb

#Print all feature classes in "BlackHills.gdb"
arcpy.env.workspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'
bh_fc = arcpy.ListFeatureClasses()
print("Feature classes in \"BlackHills\": ")
for fc in bh_fc:
    print(fc)
```

Question 1 output:

```
========== RESTART: C:/PythonPro/Exercise06/q1.py
Feature classes in "BlackHills":
streams
towers
```

Question 2 script:

```
# Lab 6 Question 2
import arcpy
import os
arcpy.env.overwriteOutput = True
wkspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'
data_folder = 'C:/PythonPro/Exercise06/Lab06Data'
arcpy.env.workspace = data_folder
raster_list = arcpy.ListRasters()
for rastr in raster_list:
    rastr_name = arcpy.da.Describe(rastr)["baseName"]
    new_rastr = os.path.join(wkspace, rastr_name)
    arcpy.management.CopyRaster(rastr, new_rastr)

# Print: Name, format, compression, # of bands, coordinate system.
arcpy.env.workspace = wkspace
bh_rast = arcpy.ListRasters()
for rastr in bh_rast:
    desc = arcpy.da.Describe(rastr)
    print("Raster Name: " + desc["baseName"])
    print("Format: " + desc["format"])
    print("Compression: " + desc["compressionType"])
    print("Number of Bands: " + str(desc["bandCount"]))
    print("Coordinate System: " + desc["spatialReference"].name + '\n')
```

Question 2 output:

```
========== RESTART: C:/PythonPro/Exercise06/q2.py =
Raster Name: landcover
Format: FGDBR
Compression: LZ77
Number of Bands: 1
Coordinate System: NAD_1983_UTM_Zone_13N

Raster Name: topo
Format: FGDBR
Compression: LZ77
Number of Bands: 1
Coordinate System: NAD_1983_UTM_Zone_13N
```

Question 3 script:

```
# Lab 6 Question 3
import arcpy
arcpy.env.overwriteOutput = True
arcpy.env.workspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'

fc = "streams"
with arcpy.da.SearchCursor(fc, ["SHAPE@LENGTH"]) as cursor:
    all_streams = []
    for row in cursor:
        all_streams.append(row[0])
    all_streams.sort(reverse=True)
    print("The lengths of the 5 longest streams are:")
    for strm in all_streams[0:5]:
        print(f"{strm} meters")
```

Question 3 output:

```
==================== RESTART: C:/PythonPro/Exercise06/q3.py
The lengths of the 5 longest streams are:
11649.859510516064 meters
9218.911796772265 meters
7790.3215826421 meters
7111.874155852162 meters
7046.046280993783 meters
```

Question 4 script:

```
# Lab 6 Question 4
import arcpy
import fileinput
arcpy.env.overwriteOutput = True
wkspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'
arcpy.env.workspace = wkspace

new_pgfc = "bh_fire"
arcpy.management.CreateFeatureclass(wkspace, new_pgfc, "Polygon", spatial_reference = "streams")
infile = "C:/PythonPro/Exercise06/Lab06Data/fireperim.txt"
with arcpy.da.InsertCursor(new_pgfc, ["SHAPE@"]) as cursor:
    array = arcpy.Array()
    for line in fileinput.input(infile):
        ID, X, Y = line.split()
        array.add(arcpy.Point(X, Y))
    cursor.insertRow([arcpy.Polygon(array)])
    fileinput.close()

fc = "bh_fire"
vertices = 0
perimeter = 0
area = 0
with arcpy.da.SearchCursor(fc, ["OID@", "SHAPE@", "SHAPE@LENGTH", "SHAPE@AREA"]) as cursor:
    for row in cursor:
        for point in row[1].getPart(0):
            vertices += 1
        perimeter += row[2]
        area += row[3]

desc = arcpy.da.Describe(fc)
units = desc["spatialReference"].linearUnitName

print(f"The Black Hills fire polygon has {vertices} vertices, a perimeter of {perimeter} {units}s, and covers {area} {units}s squared of area.")
```

Question 4 output:

```
==================== RESTART: C:/PythonPro/Exercise06/q4.py ====================
The Black Hills fire polygon has 1111 vertices, a perimeter of 33299.69860023494
Meters, and covers 20876885.457385115 Meters squared of area.
```

Question 5 script:

```
# Lab 6 Question 5
import arcpy
arcpy.env.overwriteOutput = True
arcpy.env.workspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'

min_elev = arcpy.management.GetRasterProperties("topo", "MINIMUM")
max_elev = arcpy.management.GetRasterProperties("topo", "MAXIMUM")
desc = arcpy.da.Describe('topo')
units = desc['spatialReference'].linearUnitName
print(f"The black hills field site ranges from {min_elev} {units}s to {max_elev} {units}s.")
```

Question 5 output:

```
==================== RESTART: C:/PythonPro/Exercise06/q5.py ====================
The black hills field site ranges from 954.715759277344 Meters to 1958.61889648438 Meters.
```

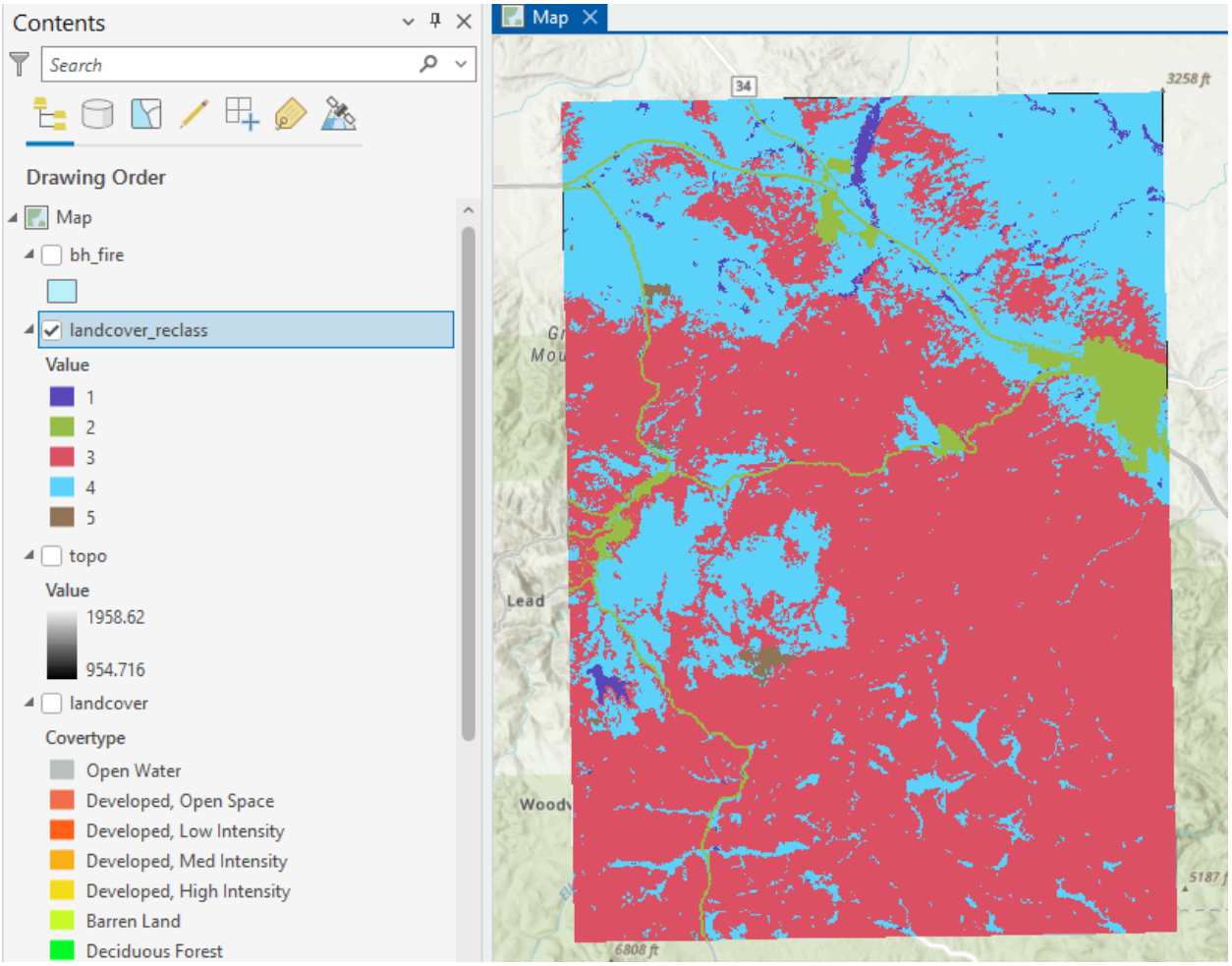
Question 6 script:

```
# Lab 6 Question 6
import arcpy
import numpy
arcpy.env.overwriteOutput = True
arcpy.env.workspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'

lc_remap = arcpy.sa.RemapValue([
    [11, 1], [90, 1], [95, 1],
    [21, 2], [22, 2], [23, 2], [24, 2],
    [41, 3], [42, 3], [43, 3],
    [52, 4], [71, 4], [81, 4], [82, 4],
    [31, 5]
])
out_recl = arcpy.sa.Reclassify('landcover', 'Value', lc_remap, 'NODATA')
out_recl.save("landcover_reclass")

r = arcpy.Raster("landcover_reclass")

# Convert raster to NumPy array
arr = arcpy.RasterToNumPyArray(r)

# Get cell size
cell_area = r.meanCellWidth * r.meanCellHeight

# Count pixels per class
unique, counts = numpy.unique(arr, return_counts=True)

desc = arcpy.da.Describe("landcover_reclass")
unit = desc['spatialReference'].linearUnitName

# Print area for each class
for val, count in zip(unique, counts):
    if val == 0: continue #skip to next loop iteration
    area = count * cell_area
    print(f"Class {val}: {area} square {unit}s")
```

Question 6 output:

```
============================ RESTART: C:/PythonPro/Exercise06/q6.py ==
Class 1: 5035500.0 square Meters
Class 2: 19443600.0 square Meters
Class 3: 354188700.0 square Meters
Class 4: 172737000.0 square Meters
Class 5: 1801800.0 square Meters
```

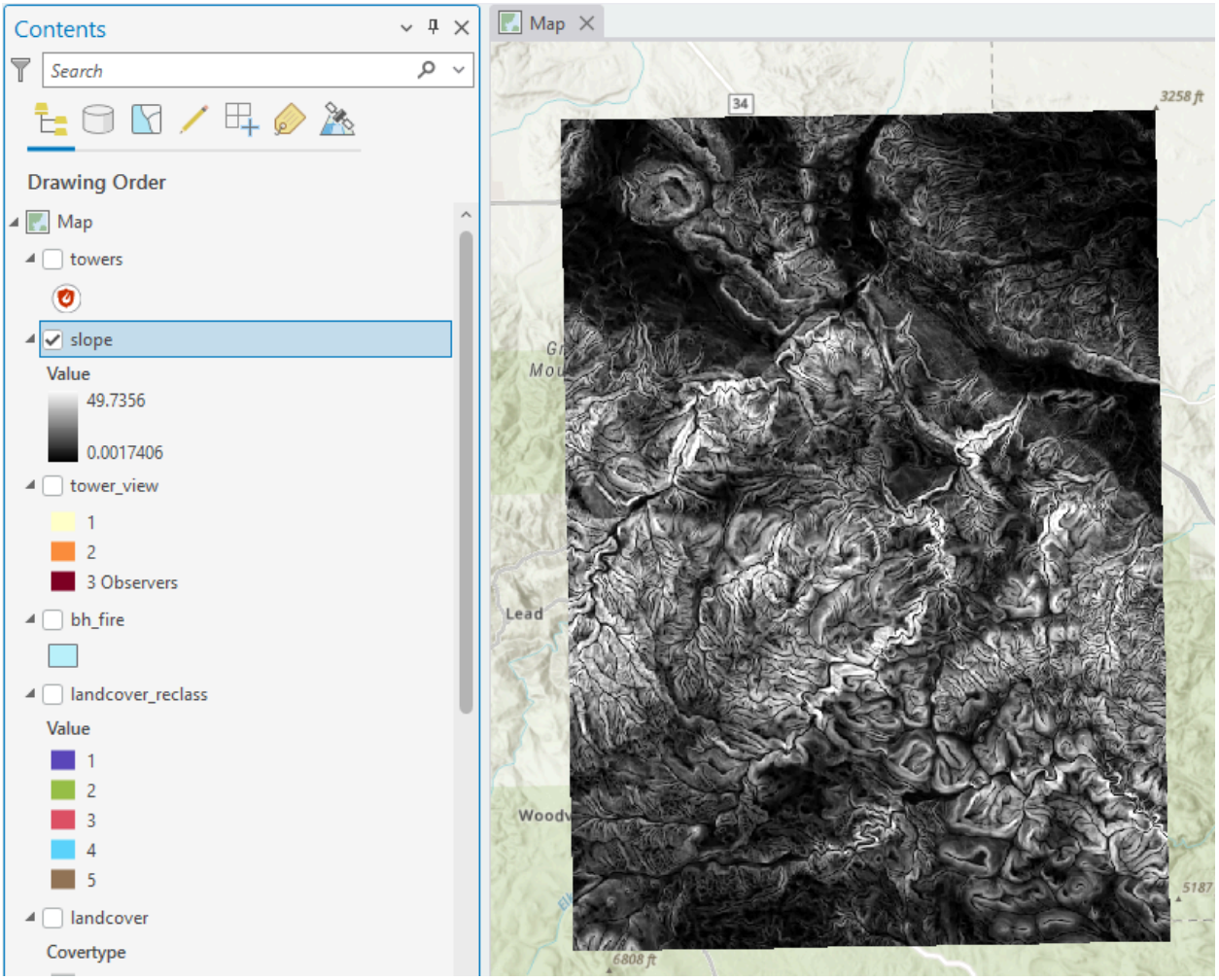
Question 6 map:



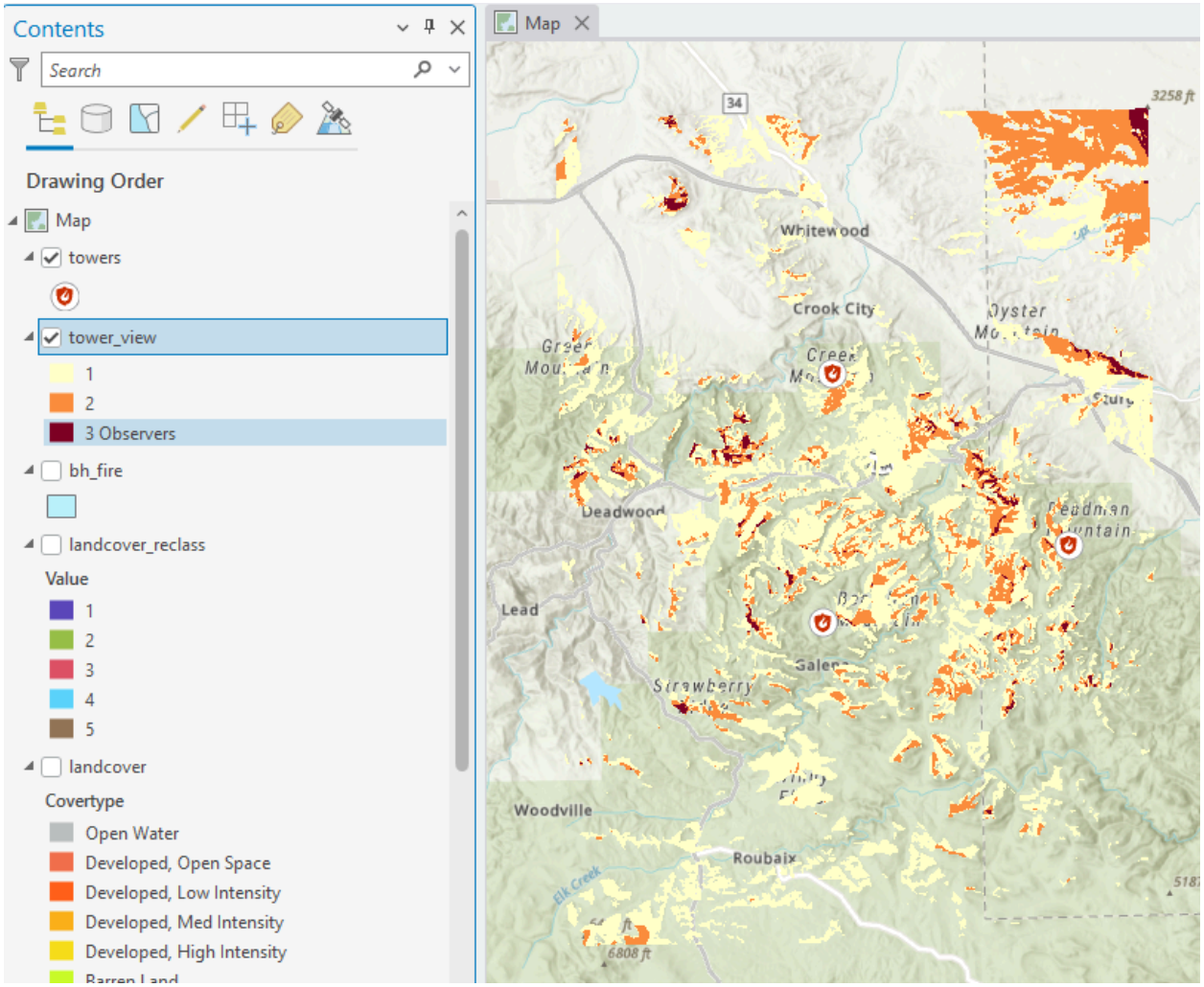
Question 7 script:

```
# Lab 6 Question 7
import arcpy
arcpy.env.overwriteOutput = True
arcpy.env.workspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'

outraster = arcpy.sa.Slope("topo")
outraster.save("slope")

outraster = arcpy.sa.Viewshed(
    in_raster="topo",
    in_observer_features="towers",
    z_factor=1,
    curvature_correction="FLAT_EARTH",
    refractivity_coefficient=0.13,
    out_agl_raster=None
)
outraster.save("tower_view")
```

Question 7 output:
Slope:



Viewshed:


Contents
Search
Drawing Order
Map
towers
tower_view
1
2
3 Observers
bh_fire
landcover_reclass
Value
1
2
3
4
5
landcover
Covertype
Open Water
Developed, Open Space
Developed, Low Intensity
Developed, Med Intensity
Developed, High Intensity
Map
34
3258 ft
Whitewood
Crook City
Deadwood
Lead
Strawberry
Woodville
Roubaix
Elk Creek
6808 ft

Question 8 script:

```
# Lab 6 Question 8 (Challenge)
import arcpy
arcpy.env.overwriteOutput = True
arcpy.env.workspace = 'C:/PythonPro/Exercise06/BlackHills.gdb'

# Get the fire perimeter geometry from the 'bh_fire' polygon
with arcpy.da.SearchCursor("bh_fire", ["SHAPE@"]) as fc_cursor:
    fire_geom = next(fc_cursor)[0]

# Get units from spatial reference for printing purposes later
desc = arcpy.da.Describe("towers")
units = desc["spatialReference"].linearUnitName

# Loop through towers and measure distance
with arcpy.da.SearchCursor("towers", ["OID@", "SHAPE@", "NAME"]) as tower_cursor:
    for row in tower_cursor:
        oid = row[0]
        tower_geom = row[1]
        distance = tower_geom.distanceTo(fire_geom)
        print(f"The {row[2]} tower is {distance:.3f} {units}s from the fire perimeter")
```

Question 8 output:

```
========== RESTART: C:/PythonPro/Exercise06/q8.py ========
The Crook Mountain tower is 3762.626 Meters from the fire perimeter
The Deadman Mountain tower is 7622.907 Meters from the fire perimeter
The Bear Den Mountain tower is 1430.276 Meters from the fire perimeter
```