Exercise 13: Cursor Objects

The cursor functions on ArcPy return in-memory objects containing one or more rows of data from a feature class or table. Cursor objects provide methods that allow you to iterate through each row in the cursor. Various types of cursors can be created depending upon need. Search cursors can be created for read-only access to data. Update cursors can be created to update or delete rows, and insert cursors can be created to insert new rows in a feature class or table.

At the end of this exercise you will have learned the following:

- Create a SearchCursor containing a read only in-memory recordset
- Use a where clause to restrict the records returned in the cursor
- Use an UpdateCursor to update records in a feature class

Step 1: Create a SearchCursor

In this step you will create a Python script that creates a SearchCursor that will be used to loop through the records of the Schools shapefile. We will apply a where clause to return only schools that are listed as Primary schools.

- Open PythonWin and create a new script.
- Insert comments including your name, date, and the purpose of this script.
- Save the script as SearchCursor.py.
- Import the ArcPy module

```
import arcpy
```

- Create a try block and set the workspace

```
>>> import arcpy
>>> try:
...     arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
...
```

- Create a search cursor object containing only Primary schools. The first argument to the SearchCursor method is the name of the feature class that we will use. The second parameter is a where clause that can be used to create a subset of records that will fill the cursor object. In this case we define our where clause as “TYPE = ‘Primary School’.

```
>>> import arcpy
>>> try:
...     arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS
Programming\Training\Data"
...     searchCurs = arcpy.SearchCursor("Schools.shp","\"TYPE\" =
\'Primary School\'")
...
```

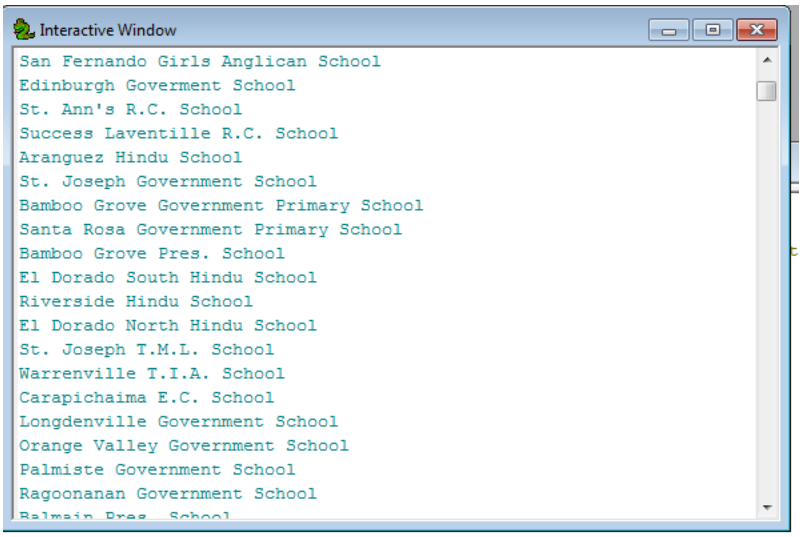
- Loop through each row in the SearchCursor and print out the name of the school. Here we are using a for loop to loop through each school returned. To access the values from a field/column on the row you can use one of two methods. The first method uses dot notation to obtain the value which is the method we have used in this case. The second method of retrieving data for a field in a row is to use the GetValue method found on the Row object. The GetValue method accepts a field name as you can see in the code below. You can use either method.

```
>>> import arcpy
>>> try:
...     arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS
Programming\Training\Data"
...     searchCurs = arcpy.SearchCursor("Schools.shp","\"TYPE\" =
\'Primary School\'")
...     for row in searchCurs:
...             print row.SCHOOL
```

- Add an except block to handle any errors.

```
>>> import arcpy
>>> try:
...     arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS
Programming\Training\Data"
...     searchCurs = arcpy.SearchCursor("Schools.shp","\"TYPE\" =
\'Primary School\'")
...     for row in searchCurs:
...             print row.SCHOOL
... except:
...     arcpy.GetMessages()
```

- Save and run the script. The script will print out the name of each Primary school



Step 2: Use UpdateCursor to Update All Rows in a Feature Class

In this step you will add a field to the Hospitals shapefile using the Add Field tool in the Data Management toolbox, and then update this field to hold the name and address of the facility using an UpdateCursor.

- Open PythonWin and create a new script.
- Insert comments including your name, date, and the purpose of this script.
- Save the script as Add_Update_Field.py.
- Import the ArcPy module

```
import arcpy
```

- Create a try block and set the workspace

```
>>> import arcpy
>>> try:
...     arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
...
```

- Add a field to the Hospitals shapefile by using the AddField tool found in the Data Management toolbox. The new field will be called 'FullAddr' and will be a text field with a length of 50 characters.

```
import arcpy
try:
    arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
    arcpy.AddField_management ("Hospitals.shp","FullAddr","TEXT", "50")
```

- Create an update cursor containing all hospitals in the hospitals shapefile

```
import arcpy
try:
    arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
    arcpy.AddField_management ("Hospitals.shp","FullAddr","TEXT", "50")
    updCursor = arcpy.UpdateCursor("Hospitals.shp")
```

- Start a for loop to loop through each of the records. Inside the for loop you'll want to pull out the name and address for each record and then combine them into a new variable called 'FullAddress' which contains the concatenation of these address variables.

```
import arcpy
try:
    arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
    arcpy.AddField_management ("Hospitals.shp","FullAddr","TEXT", "50")
    updCursor = arcpy.UpdateCursor("Hospitals.shp")
    for row in updCursor:
        name = NAME
        addr = ADDRESS
        FullAddress = name + "," + addr
```

- Call the SetValue method on the Row object. The first parameter for SetValue is the name of the field that will be updated, and the second parameter contains the value that will be inserted into the field for a particular row. To apply this new value you then need to call the updateRow method found on the UpdateCursor object you created.

```
import arcpy
try:
    arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
    arcpy.AddField_management ("Hospitals.shp","FullAddr","TEXT", "50")
    updCursor = arcpy.UpdateCursor("Hospitals.shp")
    for row in updCursor:
        name = NAME
        addr = ADDRESS
        FullAddress = name + "," + addr
        row.setValue("FullAddr",FullAddress)
        updCursor.updateRow(row)
```

- After the for loop has updated all the records you will then print a message to the user stating that the rows have been updated, and then you will need to remove the reference to the cursor.

```
import arcpy
try:
    arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
    arcpy.AddField_management ("Hospitals.shp","FullAddr","TEXT", "50")
    updCursor = arcpy.UpdateCursor("Hospitals.shp")
    for row in updCursor:
        name = NAME
        addr = ADDRESS
        FullAddress = name + "," + addr
        row.setValue("FullAddr",FullAddress)
        updCursor.updateRow(row)
    print "Finish updating"
    del updCursor
```

Note: Make sure that the ‘del updCursor’ statement is outside the while loop. You do this by making sure that the del and for statements line up. If your delete statement is inside the cursor an error will occur after the first record has been processed. You don’t want to delete the cursor until you’re done with it.

- Add an except block to handle any errors.

```
import arcpy
try:
    arcpy.env.workspace = r"C:\Users\Administrator\Desktop\GIS Programming\Training\Data"
    arcpy.AddField_management ("Hospitals.shp","FullAddr","TEXT", "50")
    updCursor = arcpy.UpdateCursor("Hospitals.shp")
    for row in updCursor:
        name = NAME
        addr = ADDRESS
        FullAddress = name + "," + addr
        row.setValue("FullAddr",FullAddress)
        updCursor.updateRow(row)
    print "Finish updating"
    del updCursor
except:
    arcpy.GetMessages()
```

- Save and run the script.
- Open ArcMap and examine the Hospitals shapefile. Open the Attribute Table and scroll to the end of the table. A new field called “FullAddr” should have been added to the table, and populated with the full address for each facility.

PROBLEM:

- The FullAddr field remained empty, explain why? (Hint row.attribute name)