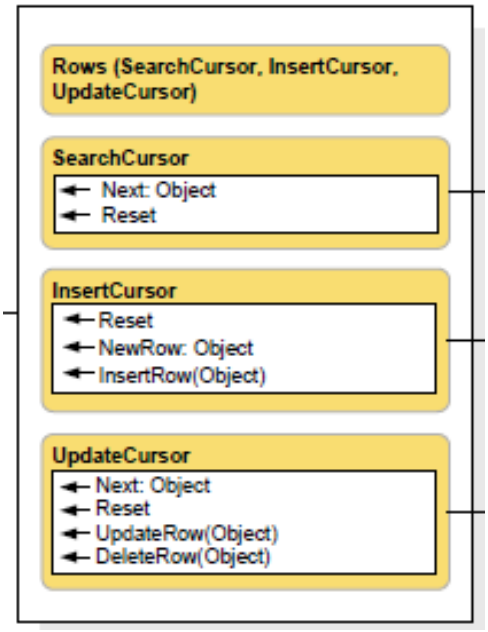
Chapter 13: Reading and updating GIS Data

- A cursor is a data access object that can be used to iterate through the set of rows to select, add or update rows in a feature class table.
- Cursors have three forms: search, insert, or update.
- Cursors are commonly used to read and update attributes.
- All updates or insertions to a table are done outside an edit session in ArcGIS.
- Changes made are permanent and cannot be undone.

Method	Explanation
deleteRow (row)	Deletes a row in the database. The row corresponding to the current position of the cursor will be deleted.
insertRow (row)	Inserts a new row into the database.
newRow ()	Creates an empty row object.
next ()	Returns the next object at the current index.
reset ()	Sets the current enumeration index (used by the next method) back to the first element.
updateRow (row)	The updateRow method can be used to update the row at the current position of an update cursor.

- There are 3 cursor functions in Arcpy. Each creates a cursor object with the same name.



- Each cursor function has methods that allow you to access Rows
- Cursors can only be navigated in a forward direction; they do not support backing up and retrieving rows that have already been retrieved or making multiple passes over data.

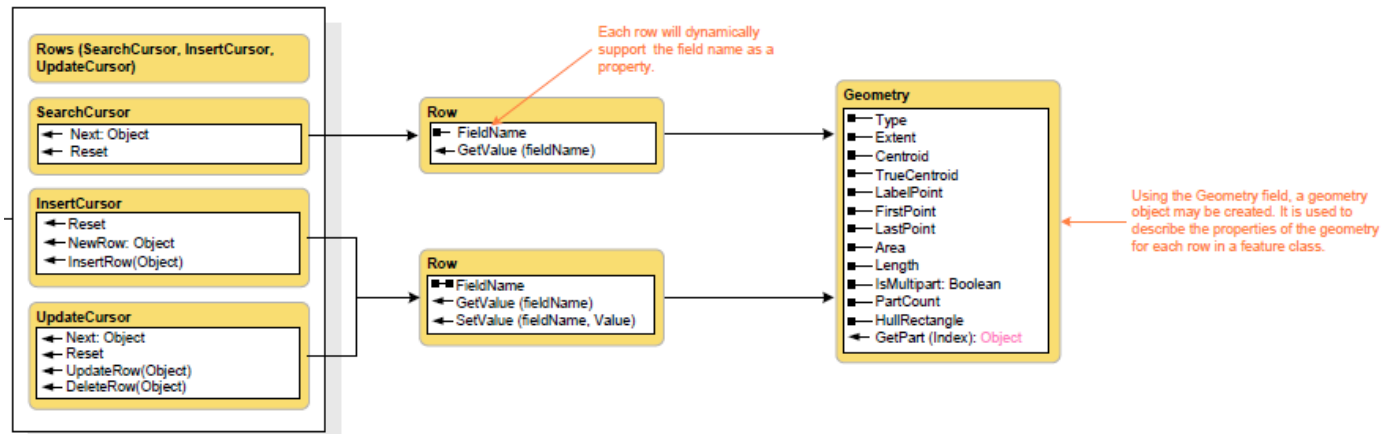
- If a script needs to make multiple passes over the data, the application must re-execute the cursor function.
- If both executions are made within the same edit session (or database transaction with the appropriate level of isolation), the application is guaranteed not to see any changes made to the data by other concurrently executing applications.

13.1 Cursor Objects

- SearchCursor, UpdateCursor, and InsertCursor functions create a cursor object (sometimes referred to as an enumeration object) that can be used to iterate through the records.
- The methods of the cursor object created by the various cursor functions vary depending on the type of cursor created.

Cursor type	Method	Effect on position
Search	`next`	Retrieves the next row object
Insert	`newRow`	Creates an empty row object
	`insertRow`	Inserts a row object into the table
	`next`	Retrieves the next row object
Update	`updateRow`	Updates the current row with a modified row object
	`deleteRow`	Removes the row from the table
	`next`	Retrieves the next row object

- The cursor object model diagram



- Arcpy contains methods for creating each of the various cursor objects.
- The SearchCursor, InsertCursor and UpdateCursor returns corresponding objects
- Each cursor object has methods which give access to row objects which give access to field names and values
- In addition, the shape field returns an instance of the geometry object which contains detailed geometry information for each row.

13.2 InsertCursor()

- Creates a new cursor object used for adding new rows to a table, feature class or shapefile
- To create new empty rows, use the NewRow method under InsertCursor
    - Cursor.NewRow()
- The NewRow method returns an instance of the row object which can then be used to populate the row with new values through the setValue method
    - Row.setValue(fieldname, value)

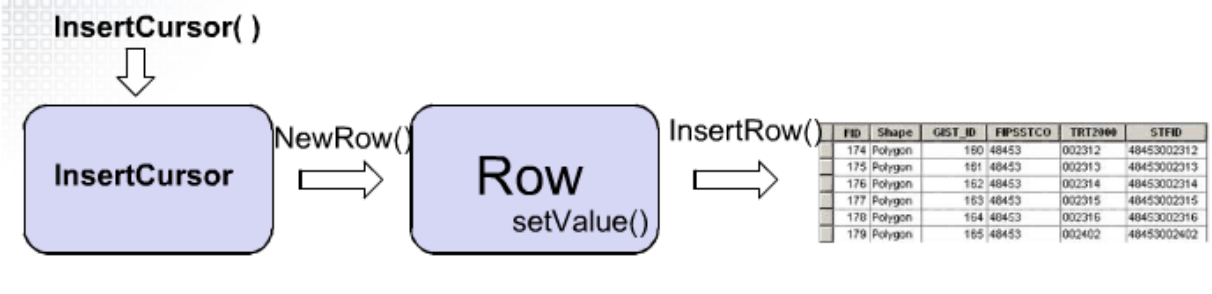
Syntax

InsertCursor (dataset, {spatial_reference})

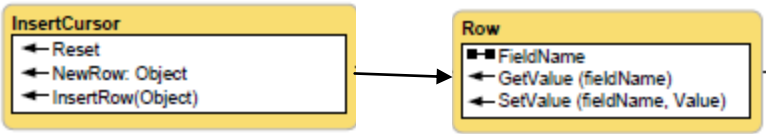
Parameter	Explanation	Data Type
dataset	The table, feature class, or shapefile into which rows will be inserted.	String
spatial_reference	The Spatial Reference of the feature class, shapefile, or table. This object may be obtained from the arcpy Describe function.	Object

Return Value

Data Type	Explanation
Object	Returns a Cursor object against the specified feature class, shapefile, or table.



FID	Shape	GIST_ID	FIPSSTCO	TRT2000	STFID
174	Polygon	180	48453	002312	48453002312
175	Polygon	181	48453	002313	48453002313
176	Polygon	182	48453	002314	48453002314
177	Polygon	183	48453	002315	48453002315
178	Polygon	184	48453	002316	48453002316
179	Polygon	185	48453	002402	48453002402



Example

Adds a new row to a shapefile with new attribute information

```
import arcpy
insertCurs = arcpy.InsertCursor("C:/temp/Census_Tracts.shp")
row = insertCurs.newRow()
row.EditedBy = "John Doe"
row.TractID = "204.05"
insertCurs.insertRow(row)
del row
del insertCurs
```

Create an InsertCursor object

Create a Row and edit attributes

Insert the Row into FeatureClass

- The row will be added to the bottom of the attribute table
- Task:
    - Try the above with one of your shapefiles.
- Example: insertCursor2.py

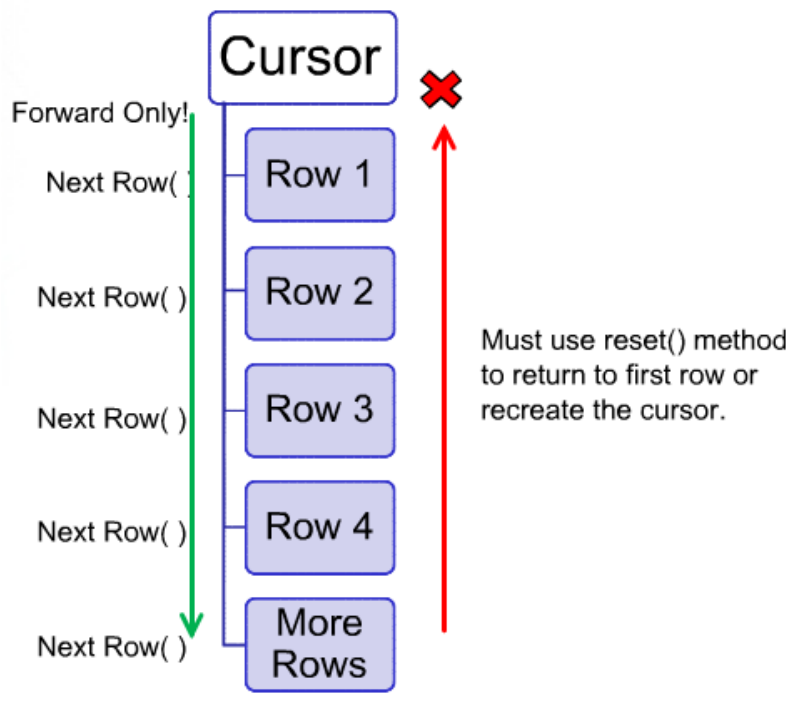
```
import arcpy
# Create insert cursor for table
#
rows = arcpy.InsertCursor(r"C:\Users\Me\Desktop\GIS Programming\Training\Data\Hospitals.shp")
x = 1

# Create 25 new rows. Set the initial row ID and distance values
#
while x <= 25:
    row = rows.newRow()
    row.FID_ = x
    row.AREA = 100
    rows.insertRow(row)
    x = x + 1
```

Task:

- What does the above Python script do?

13.3 SearchCursor() function

- Used to return a searchCursor enumeration object
- This object can only be used to iterate through a set of rows for READ ONLY purpose
- No insertions, deletions or updates can occur through this object
- An option WHERE clause can be set to limit the row returned
- Iteration through the rows occurs through the Next and Reset methods
- Next returns the next row object in the list
- No method for moving backward through the cursor one a row has been visited using the Next method
- Initially, the cursor pointer is just above the first row
- To re-visit a row, you need to call the Reset method which resets the pointer to the first element
    - You can always create another cursor



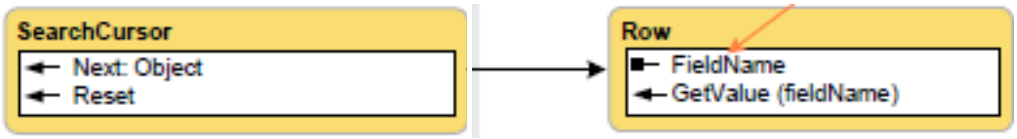
Syntax

SearchCursor (dataset, {where_clause}, {spatial_reference}, {fields}, {sort_fields})

Parameter	Explanation	Data Type
dataset	The feature class, shapefile, or table containing the rows to be searched.	String
where_clause	An optional expression that limits the rows returned in the cursor. For more information on WHERE clauses and SQL statements, see About building an SQL expression.	String
spatial_reference	The Spatial Reference of the feature class.	Object
fields	The fields to be included in the cursor. By default, all fields are included.	String
sort_fields	Fields used to sort the rows in the cursor. Ascending and descending order for each field is denoted by A and D.	String

Return Value

Data Type	Explanation
Object	A Cursor object that can hand out row objects.



Example

List field contents for Counties.shp. Cursor sorted by State Name and Population.

```
import arcpy

# Open a searchcursor
#  Input: C:/Data/Counties.shp
#  FieldList: NAME; STATE_NAME; POP2000
#  SortFields: STATE_NAME A; POP2000 D
#
rows = arcpy.SearchCursor("C:/Data/Counties.shp", "", "", "NAME; STATE_NAME; POP2000",
                          "STATE_NAME A; POP2000 D")
currentState = ""

# Iterate through the rows in the cursor
#
for row in rows:
    if currentState != row.STATE_NAME:
        currentState = row.STATE_NAME

    # Print out the state name, county, and population
    #
    print "State: %s, County: %s, population: %i" % \
            (row.STATE_NAME, row.NAME, row.POP2000)
```

13.4 UpdateCursor()

- Can be used to update or delete rows in a Table feature class
- The cursor places a lock on the data that will remain until either the script completes or the update cursor object is deleted.
- Its good practice to delete the create cursor after it has been used to explicitly remove the lock on the data.
- Can apply a WHERE clause to filter the returned data
- Call updateRow() method to save changes at the end of update operations

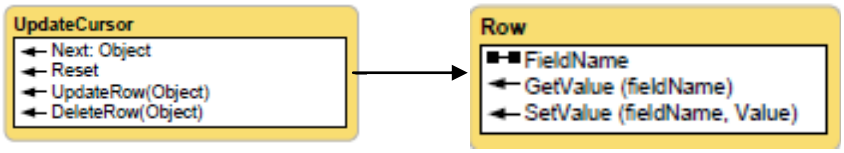
Syntax

UpdateCursor (dataset, {where_clause}, {spatial_reference}, {fields}, {sort_fields})

Parameter	Explanation	Data Type
dataset	The feature class, shapefile, or table containing the rows to be updated or deleted.	String
where_clause	An optional expression that limits the rows returned in the cursor. For more information on WHERE clauses and SQL statements, see About building an SQL expression.	String
spatial_reference	The Spatial Reference of the feature class.	Object
fields [fields,...]	The fields to be included in the cursor. By default, all fields are included.	String
sort_fields	Fields used to sort the rows in the cursor. Ascending and descending order for each field is denoted by A and D.	String

Return Value

Data Type	Explanation
Object	A Cursor object that can hand out row objects.

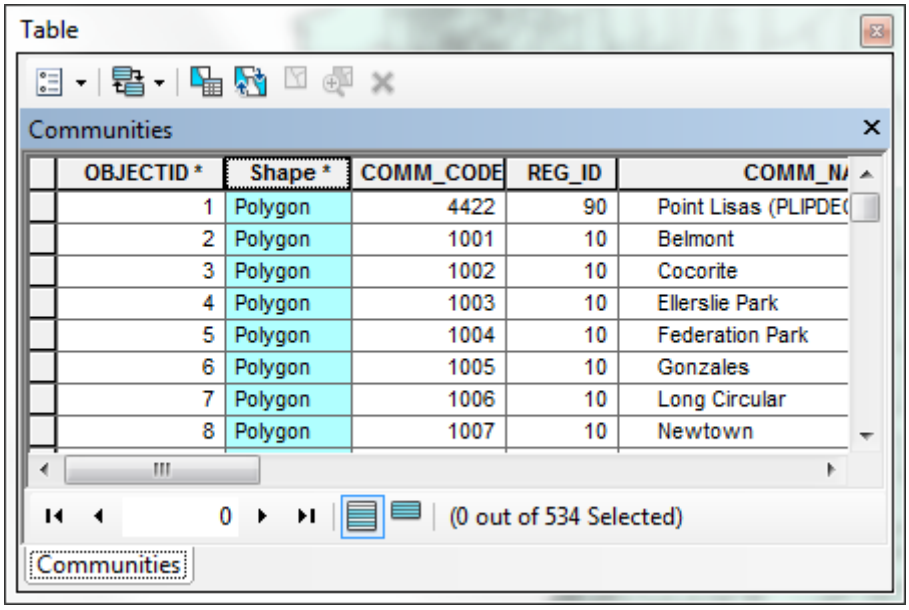


Example

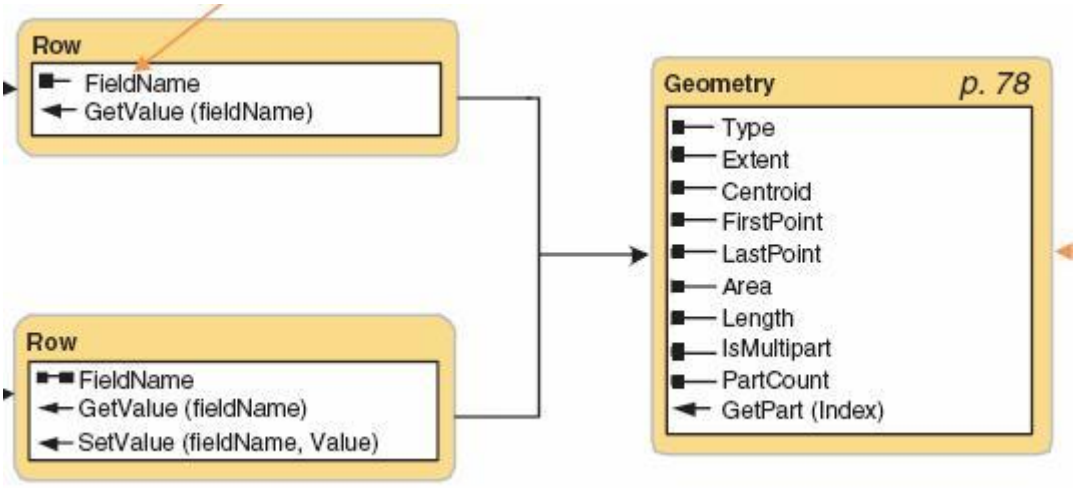
```
import arcpy

# Create update cursor for feature class
#                          Create the update cursor
rows = arcpy.UpdateCursor("D:/St_Johns/data.gdb/roads")

# Update the field used in buffer so the distance is based on the road
# type. Road type is either 1, 2, 3 or 4. Distance is in meters.
#          Loop through records in cursor
for row in rows:
    # Fields from the table can be dynamically accessed from the row objec
    #   Here fields named BUFFER_DISTANCE and ROAD_TYPE are used
    row.BUFFER_DISTANCE = row.ROAD_TYPE * 100
    rows.updateRow(row)
                    Update values in the feature class or table
# Delete cursor and row objects to remove locks on the data
#
del row
del rows    Remove locks
```

13.5 Geometry class

- Each row object in a FeatureClass has an associated Geometry object containing the geometric content for the row. This is more commonly known as the “Shape”field.



- When you access the “Shape” field from a Row object a Geometry object is returned, and contains a number of read only properties that give you geographic information about the current row. Notice the relationship between Row objects and Geometry objects in the figure below.



- Some of the more commonly used properties include:

- Type – data type (point, polyline, polygon)
- Extent – geographic extent (bounding box) of the feature
- Area – area of a polygon feature
- Length – length of a polyline feature

13.6 Reading geometries

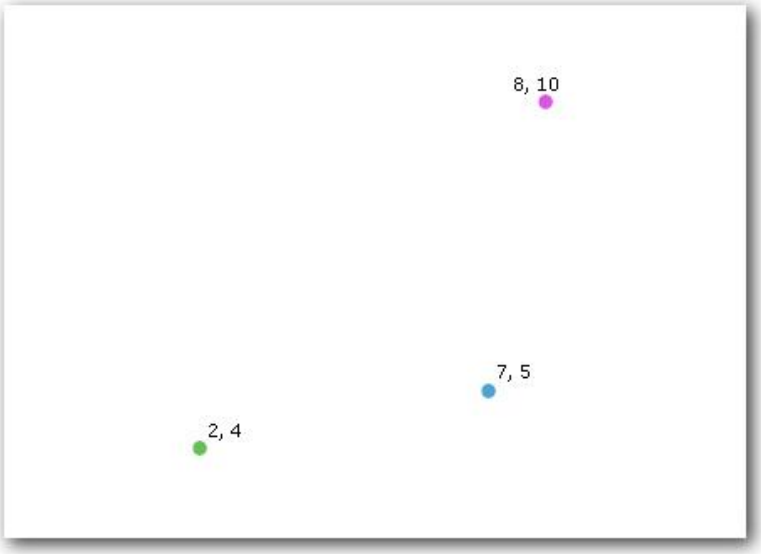
- You can read the geometries of each feature in a feature class through the geometry object
- Each feature contain a sets of points which define the vertices of a polygon or a polyline or a single coordinate for a point feature
- An array of point objects is returned for polygons and polylines, while a single point object is returned for a point feature class

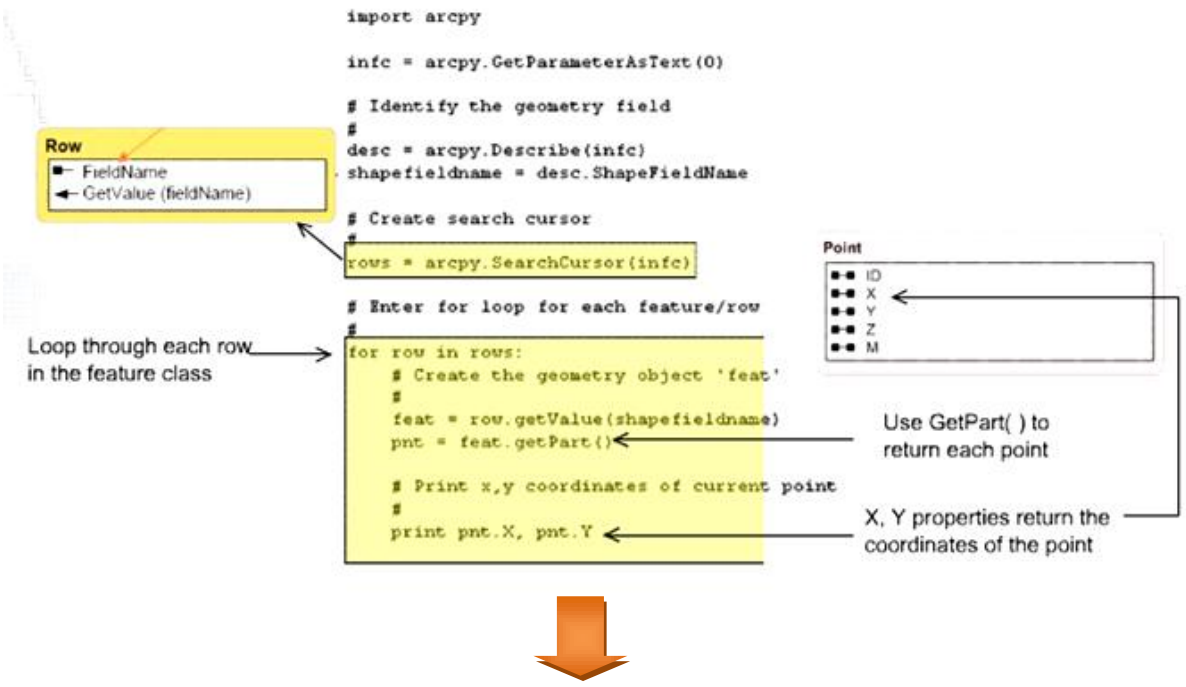
Property	Explanation
ID	The shape ID of the point
X	The horizontal coordinate of the point
Y	The vertical coordinate of the point
Z	The elevation value of the point
M	The measure value of the point

- The method for reading geometry varies by feature class type
- The geometry object's `partCount` property returns the number of parts for a feature.
- The `getPart` method returns an array of point objects for a particular part of the geometry if an index is specified. If an index is not specified, an array containing an array of point objects for each geometry part is returned.

Geometry.GetPart()

- If a polygon contains holes, it will consist of a number of rings. The array of point objects returned for a polygon contains the points for the exterior ring and all inner rings.
- The exterior ring is always returned first, followed by inner rings, with null point objects as the separator between rings. Whenever a script is reading coordinates for polygons in a geodatabase or shapefile, it should contain logic for handling inner rings if this information is required by the script; otherwise, only the exterior ring is read.
- A ring is a closed path that defines a two-dimensional area. A valid ring consists of a valid path, such that the from and to points of the ring have the same x,y coordinates. A clockwise ring is an exterior ring, and a counterclockwise ring defines an interior ring.





2.0 4.0
8.0 10.0
7.0 5.0

- http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Reading_geometries/002z0000001t000000/
- Polylines and polygons can have multiparts
- Geometry.PartCount returns the number of parts per feature
- You can then use Geometry.GetPart for each part in the feature to loop through and pull out the coordinate information
- GetPartIndex(index) or GetPart()
    - Use index to return a specific part as an array
    - No index specified returns an array containing an array of points
- Using insert and update cursors, scripts can create new features in a feature class or update existing ones.
- Process
    - Create a point object
    - Populate properties
    - Place into array (not necessary for point feature class)
    - Get a row (update) or create a new row (insert)
    - Set Row.shape property equal to the array or point
    - Use InsertRow or UpdateRow to write feature to the feature class
- See http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Writing_geometries/002z0000001v000000/ for an example.
-