GIS 540 Exam I

Course number: GIS 540

Test ID: exam I

Instructor: Dr. Tateosian

Time Limit: 120 minutes

Materials allowed: One hand written page of notes (front and back), scrap paper (students may not take note page or scrap paper away from the exam).

General instructions: **Write clearly**. Use complete sentences for short answer questions. Write name and unity login id (e.g. jkrowlin) on each page. Return all pages of the exam, note page, *and* scrap paper to the proctor. Do not discuss the exam with other students before the exams are returned. Write clearly and make sure that the difference between upper and lower case letters is easily distinguishable.

Total points available _____
Total points deducted _____
Total points earned _____

Name_____ NCSU Login ID _____

1. Given the Python code statements in the left column, specify if the statement in the middle column is true or false.

Python code	**Statement**	**True or False?**
species = 'trout'	species is a *string literal*.	
x = 5.0	x is an integer type variable.	
"256" > "2400"	The statement evaluates to True.	
str(5) == '5'	The statement evaluates to True.	
foo = "GIS" foo[:2]==foo[2]	The second line of code evaluates to True.	

2. Label each item in the list as **M** for built-in module, **F** for built-in function, **C** for built-in constant, **E** for built-in exception, **K** for Python keyword, or **N** for none of the other choices. *There are three of each.*

____ IndexError	____ os	____ max	____ for	____ arcpy
____ sys	____ SyntaxError	____ by	____ in	____ TypeError
____ False	____ elseif	____ math	____ None	
____ type	____ True	____ and	____ range	

3. Write a Python FOR-loop equivalent to the following WHILE-loop

WHILE-loop	FOR-loop

x = 1
while x < 500:
  print x
  x = x + 1	

4. Beside each code fragment in the table below, give the output. If the code would cause an error, write *ERROR* and give a *brief explanation*.

Python code	**Output or cause of error**
count = 40 name = 'Toucan' **print** 'There are' + count + name	
x = (8**2) – (5/10) + (5.0/10) **print** x	
y = 2 – 5 * 6 **print** y	
fieldName = 'rock age' fieldName.upper( ) **print** fieldName	
myList = ['C','A','B'] myList.sort() **print** myList	
myList = range(4, 12, 2) **print** myList	
print min(1, 2, 3) min = 5 **print** min(1, 2, 3)	

theTime = "10:33:06" #HH:MM:SS
timeParts = theTime.split(":")
print timeParts
print timeParts[3]	

x = -5
if x > 0 or 1:
  print 'IN'
else:
  print 'OUT'	

5. Under each arcpy listing method in the table below, give the list contents. If the list is empty, write EMPTY and give a brief explanation. Assume the user passed in the following arguments:

   "C:/dataGIS" *foo

   Assume the following code has been executed:

   import arcpy, sys

   arcpy.env.workspace = sys.argv[1]

L1 = arcpy.ListFeatureClasses("*anyon*")
L2 = arcpy.ListRasters("sys.argv[2]")
L3 = arcpy.ListRasters("Out*", "TIF")
L4 = arcpy.ListFeatureClasses("*", "Point")

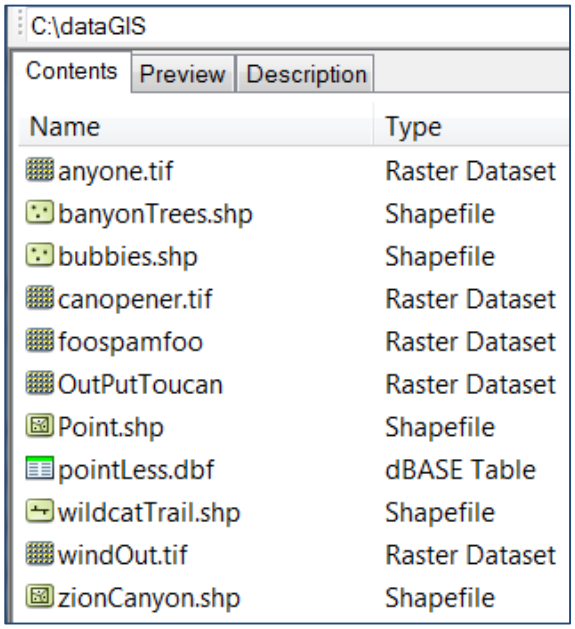

Figure 1: Contents of C:\dataGIS in ArcCatalog.

6. Write a script which prints the names of all the fields in a given input Shapefile. The script has been started in the box below.

1	import arcpy, sys
2	inputFile = sys.argv[1]
3	
4	
5	

7. Write a script which takes a year as input. Print 'CURRENT', if the input is this year (2014), 'RECENT' if the input is within the 5 years before this one (2008 through 2013), and 'TARDIS' otherwise. Also, list 3 or more sample inputs that would test the code thoroughly.

8. Answer the following questions about the traceback error shown in the box below:
    a. Which line number of 'findAvg.py' did the error occur on?

    b. What is the name of the exception?

    c. The line of code that triggered the error is shown in bold. Use variable names in this line of code to explain why the exception was thrown.

```
Traceback (most recent call last):
File "C:\Python27\ArcGIS10.2\Lib\site-packages\scriptutils.py", line 326, in RunScript
   exec codeObject in __main__.__dict__
 File "C:\sampleScripts\findAvg.py", line 8, in <module>
val = maxv - minv/rangeVal
ZeroDivisionError: integer division or modulo by zero
```

Name_____ NCSU Login ID _____

9. Identify the six mistakes in this pseudocode. Mistakes can pertain to logic, syntax, or efficiency. Line numbers for lines that have mistakes are given below (Some lines have 2 mistakes). CLEARLY show how to correct the mistakes *inside the code box.* Then, in parts 1)-7) below the box, explain the corrections.

1	GET a list of shapefiles from workspace
2	
3	DETERMINE the geometry of the current shapefile
4	IF the geometry is polygon then
5	CALCULATE the area of the polygonal lake in square feet.
6	COMPUTE buffer of the polygonal lake.
7	PRINT area of lake
8	IF the geometry is line
9	CALCULATE the length of the linear river in feet.
10	COMPUTE buffer of the linear river.
11	PRINT length of river
12	
13	

1) **Line 2 (something is missing!)**
2) **Line 4**
3) **Line 8**
4) **Line 8**
5) **Line 11**
6) **Line 12 (something is missing!)**
7) **Line 12 (something is missing!)**

10. The script ‘polygonNeighbors_nonBatch.py’ in the first box below creates a table with statistics pertaining to polygon contiguity for polygons in the ‘C:/data/COVER63p.shp’ file. Rewrite this as ‘polygonNeighbors_Batch.py’ in the second box (where the script is started for you). The batch version should call the Polygon Neighbor (Analysis) on all the polygon Shapefiles in the ‘C:/data’ directory whose names start with “COVER”.

```
1  # polygonNeighbors_nonBatch.py
2  import arcpy, os
3  arcpy.env.workspace = 'C:/data'
4  inputFile = 'COVER63p.shp'
5  outputFile = os.path.splitext(inputFile)[0] + 'PN.dbf'
6  arcpy.PolygonNeighbors_analysis(inputFile, outputFile)
7  print '{0} created.'.format(outputFile)
```

```
1  # polygonNeighbors_Batch.py
2  import arcpy, os
3  arcpy.env.workspace = 'C:/data'
4
5
6
7
8      print '{0} created.'.format(outputFile)
```

Name_____ NCSU Login ID _____

11. The screen shot in Figure 2 shows a debugging session. Use the information in the Watch Window, the current position of the cursor, and the code you can see to answer the following questions.

    a. What is the name of the file that was input into the script?

    b. Which line number is the cursor currently pointing to?

    c. Starting at the current cursor position, when the user presses the 'Step Over' button 10 times, which sound(s) will play?

    d. Give an example (if possible) of an input that won't result in any sound being played. Explain your answer.

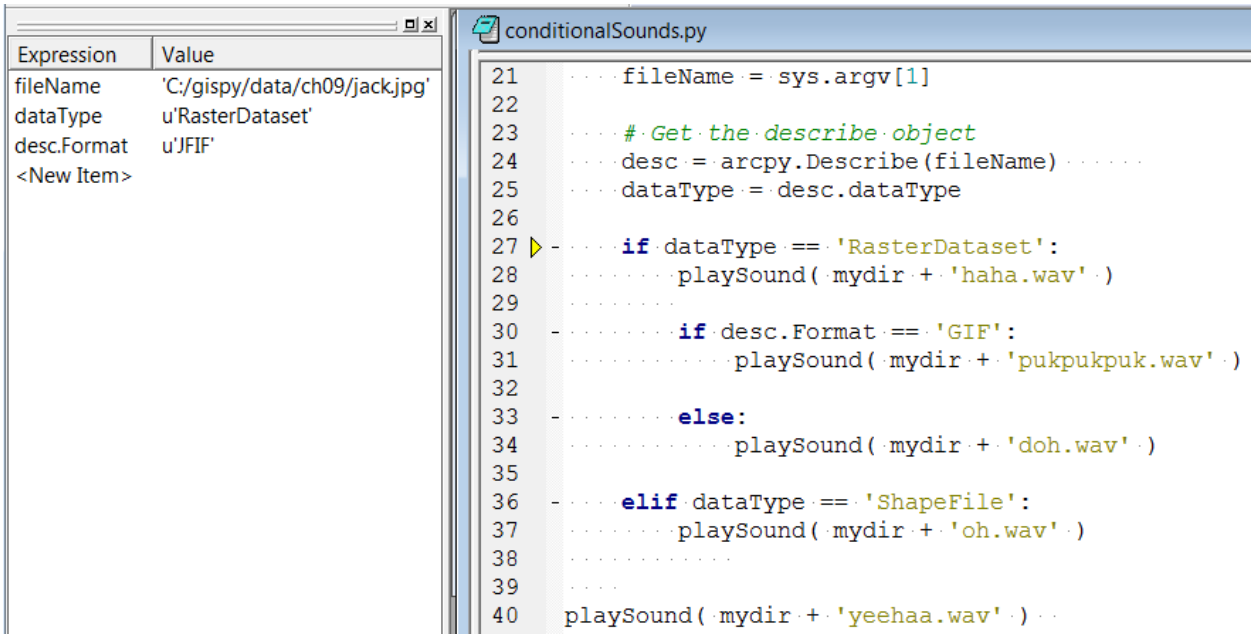

Figure 2: PythonWin debugging session.

12. Rewrite the workflow described below in terms of pseudocode. Use appropriate **structural components** (sequences, decision-making, and repetition), **key words** and **indentation**. Use each of the pseudocode key words, GET, SET, CALCULATE, FOR, IF, THEN, CALL, ENDIF, and ENDFOR, one or more times in your solution.

Get a workspace from the user and get a list of the files in that workspace. Then check the modification time of each file and delete any files that have not been modified for 5 years.

13. The following script is meant to use a WHILE-loop to create 1 mile, 2 mile, 3 mile, 4 mile, and 5 mile buffer output files using the Buffer (Analysis) tool, but at the moment, it has 10 mistakes. Line numbers with mistakes are given (Some lines have 2 mistakes). CLEARLY show how to correct the mistakes *inside the code box.* Then, in parts 1)-10) below the box, explain what would happen if the mistake were not fixed.

```
1  import arcpy
2  inputFile = sys.argv[0]
3
4  while count <= 6:
5     distance = count + "miles"
6     output = inputFile[:-4] + 'buff.shp'
7     arcpy.Buffer(inputFile, output, distance)
8     print "Output created."
9  count = count + 2
```

1) Line 1

2) Line 2

3) Line 3

4) Line 4

5) Line 5

6) Line 5

7) Line 6

8) Line 7

9) Line 9

10) Line 9