Problem Solving and Programming

Assignment 2 – 10% - Due Sunday Nov 29 (Week 12) at 11:30pm

This is an individual submission and assessment. It is not to be completed in groups.

Learning Outcomes

- Use strings and lists appropriately for the storage, manipulation and processing of text and numeric data
- Use structured programming constructs, such as functions, to write modular programs
- Write data to and read data from files
- Design and use object-oriented classes for modeling objects specific to the domain of a problem or application.
- Solve simple programming problems in a spatial context

- Online submission time will determine late submissions. If the assessment is not submitted by the due date and time, a penalty of 10% of the maximum possible assessment value per day will be assigned starting after the due date and time. This penalty applies for each 24-hour period, including weekends. After 120 hours (5 days) submissions will not be accepted.
- Self-assessment marking scheme will be posted 5 days after due date and time. Self-assessment must be submitted in order to receive value for the self-assessed portion of the assignment.

Deliverables 1, 2 and 3 each require a single Python source code file (script file). Deliverable 4 requires a single Python source code file, your MyPSPTools toolbox with the specified script tool as well as Screen captures. Deliverables 5 to 8 require Screen Captures and answers in your report.

Create a single Word file for your report.

- Include your name, the date and the assignment #
- Include all your screen captures in your report. Do not submit individual screen captures as separate files.
- Make sure your screen captures are legible – it is preferred to only capture the relevant portion of your monitor (for example using the Snipping Tool) instead of your entire monitor.
- Include your screen captures at 100% whenever possible so they are very legible.
- When scripts are asked to be included in the report, copy and paste them into your report as text. Turn off automatic formatting. Ensure the indentation is appropriate for the interpretation of Python scripts.
- Only submit your individual Python source code files when specifically asked.
- Include comments in your code, as described in class, and use meaningful variable names

---

Deliverable 1: 6 marks

- Implementation in Python 3.6 of your version 2.3 whale mapping application with the following 1) an input loop, a processing loop and an output loop with input and calculated values in lists 2) the main program placed in a main function that is only called when the script is run as a standalone script 3) a programmer-defined function that receives the inputs as parameters, calculates and returns the speeed of sound and 4) a programmer-defined function that receives inputs as parameters, determines and returns the angle of the whale. All inputs to the program are entered on the screen. All outputs are displayed on the screen in tabular format aligned in columns, formatted to 2 decimal places and decimal places aligned. This version of the program should not use files. Submit a Python source code file and include comments in the code, as described in class.

Deliverable 2: 6 marks

- Implementation in Python 3.6 of your version 3.0 whale mapping application using an input loop, a processing loop and an output loop with input values read from the whaledata.csv text file, and both input and calculated values written to a csv text file, formatted as a table. The application should include a main function, a function for calculating speed of sound and a function for calculating angle of the whale. The application should be a modification of the 2.3 version of the program. Submit a Python source code file and include comments in the code, as described in class.

Deliverable 3: 6 marks

1. Submit the answers, in your report, to these two questions for the *AddGeometryAttributes* tool:
   a. What is the syntax? (not a screen capture)
   b. What are the required parameters? Include a description (not a screen capture)
   c. What are the optional parameters? Include a description (not a screen capture)
   d. Which parameters have defaults and what are their defaults? (not a screen capture)
2. Write a Python script in VS Code that uses the *AddGeometryAttributes* tool, to add the perimeter (in kilometers) and the centroid attributes to lakes.shp in your AutomationData folder (Week10). Have the tool calculate the perimeters using the Canada Lambert Conformal Conic coordinate system. Include comments in the code. Submit a Screen Capture of the code and results in your report.

Deliverable 4: 4 marks

- Your MyPSPTools toolbox with the CentralFeatureModel and the CentralFeatureScript tool from Week 11 ArcPyGeoprocessing. Include the python script file for the CentralFeatureScript tool as a separate file.
- The CentralFeatureScript tool must have metadata for the help (Summary, tags, Dialog Explanation and Scripting Explanation for each parameter, your credits, and, optionally, a Thumbnail).
- Submit screen captures in your report of the Geoprocessing Results window showing successful completion of the script when run from the Script tool as well as the messages displayed.

Deliverable 5: 6 marks

- You have been supplied an ArcGIS Project file named Question5.aprx that uses the MapExampleData from Week 9's Map Scripting with ArcPy exercise. Place the MapExampleData from Week 9 in the same folder as the Question5.aprx and connect the layers in the map to the Europe Ecities and country feature classes in the countriesGDB.gdb geodatabase. The project file contains a layout.
- Write a Python script that:
  a. Moves the Ecities layer above the country layer
  b. Adds the mjrivers feature class in the countriesGDB.gdb geodatabase to the map between the Ecities layer and the country layer. If you need to move it, then you may use insert to add the layer again into the correct position and delete the original layer.
  c. Changes the symbology of the cities layer to use Equal Interval method, 4 classes and the POP_RANK field instead of the PORT_ID field , and for a bonus, a named cross symbol.
  d. Changes the title to say "Population Ranks in Europe"
  e. Moves the legend in line with the left side of the map
  f. Adds your name to the credits
  g. Exports the layout to a pdf with your login name followed by Europe

  You may hard code the values for the variables

  Include comments in the code, as described in class. Submit a Screen Capture of the code and results in your report.

Deliverable 6: 4 marks

Referring to the Python String Manipulation Lab exercise and using the same data (week 6), use Calculate Field in ArcGIS Pro to populate the "StrtDirect" field with the direction of each street address (eg. North). For those street addresses that do not have a direction, your code should assign an empty string (""). Export the working solution to a .cal file.

- Submit a screen capture of the contents of the .cal file and a screen capture of the Calculate Field window showing the code in the Code block and the Expression box.
- Include a screen capture of the Geoprocessing History window showing successful running of Calculate Field with your code.
- Also include a screen capture of the table, after running Calculate Field with your code, showing the "StrtDirect" field for several records in the table, including both records that have a direction and records that do not have a direction.

Deliverable 7: 4 marks

- Add Python statements below each comment to sequentially do the tasks specified in the comments

```
# Program to display a menu of restaurant options and get the user's choice
# A dictionary of menu categories and options for each category will be used
# Task 1: Create a variable named category and assign the variable a list containing a
# minimum of 3 menu categories (breakfast, lunch, dinner, snack, …)

# Task 2: Create a variable named categoryOptions and assign the variable
# lists of options for each category. (eg. the options for breakfast might be
# cereal, porridge, pancakes, bacon, eggs, ….). This variable will hold a list of lists.
# The number of options does not have to be the same for each category,
# but there should be more than one option for each category

# Task 3: Create an empty dictionary

# Task 4: Create a loop that uses the variable category to add the items in the
# category list as keys to the dictionary and the items in the categoryOptions list as
# values for each key (in the same order). For example, the key breakfast would have
# a value that is a list containing cereal, porridge, pancakes, bacon, eggs

# Task 4: Use a loop and the dictionary to display the menu categories

# Task 5: Ask the user which category they would like to order from and get their
# reply. Display the user's choice of menu category

# Task 6: Based on the user's choice of menu category, use a loop and the dictionary
```

```
# to display the category options for the user's choice

# Task 7: Ask the user which category option they would like to order and get their
# reply.  Display the user's choice of category option
```

- Submit a Screen Capture of the code and results in your report.
- Bonus: Modify this program to add the cost of each category option to the dictionary as a list and once the user choses an option, display the cost for that option

Deliverable 8: 4 marks

- The following code segment creates a Lake class

```
class Lake(object):

    def __init__(self):
        self.name = name

    def SDIndex(self):
```

- Modify the Lake class definition as instructed below
  a. Add an area property and a perimeter property to the Lake class
  b. Allow the lake object to be given a name, area and perimeter when the Lake object is created (instantiated).
  c. Complete the SDIndex method to calculate and return the shoreline development index for the lake. The method should use the area and perimeter properties of the lake object. The formula for calculating shoreline development index is

$$\frac{L}{2(\sqrt{\pi A})}$$

where L = shoreline length (perimeter)
A = lake surface area

- Create a short Python program that does the following, in the order listed
  a. Obtain's the lake's name, area and perimeter from the user
  b. Create's the lake object with the name specified by the user and assigns the lake object the surface area and perimeter specified by the user
  c. Determines the lake's shoreline development index using the SDIndex method
  d. Determines whether the lake is a round lake or not. A round lake has a shoreline development index of 1.
  e. Displays the lake's name, shoreline development index and whether it is a round lake or not

- Submit a Screen Capture of the code and results in your report.