

Name: _____ Partner: _____

Python Activity 39: Object Persistence

Learning Objectives

Students will be able to:

Content:

- Define **pickling**
- Identify relevant circumstances for pickling
- Explain the parameters necessary to pickle

Process:

- Write code that pickles python objects
- Write code that loads pickled python objects.

Prior Knowledge

- Python concepts from Activities 1-20, file reading & writing

Critical Thinking Questions:

1. Examine the code in the programmer's terminal below.

```
Terminal  
0 -> python3  
1 >>> age = {'dizzy': 7, 'pixel': 1, 'tally': 2}  
2 >>> exit()  
3 -> python3  
4 >>> age['dizzy']
```

- a. What is stored in `age['dizzy']` after line 1?

- b. What does the programmer want to happen when line 4 is executed?

- c. What will happen when line 4 is executed?

- d. Write code to insert prior to line 4 to achieve the programmer's intent from question 1b.

- e. If you could invent a feature for python to avoid having to repeat your response in 1d., what might that feature be?

FYI: *Pickling* allows us to store objects by converting them to a byte stream for use later, much like placing a cucumber in a salt brine allows us to enjoy the pickle at a later time.

2. Answer the following questions by examining this code from Terminal:

```
0 -> python3
1 >>> age = {'dizzy': 7, 'pixel': 1, 'tally': 2}
2 >>> import pickle
3 >>> pickle.dump(age, open('save.pickle', 'wb'))
4 >>> exit()
5 -> python3
6 >>> import pickle
7 >>> newage = pickle.load( open('save.pickle', 'rb') )
8 >>> newage['dizzy']
9 7
```

a. What new code does the programmer add before exiting python3 this time?

b. What does the '7' on line 9 refer to?

c. What would happen if we typed `newage['tally']`?

d. What might be stored in the `newage` variable?

e. What happened in the above code?

3. Below is the pickle-specific code extracted from the lines above.

```
2 >>> import pickle
3 >>> pickle.dump(age, open('save.pickle', 'wb'))
7 >>> newage = pickle.load( open('save.pickle', 'rb') )
```

a. We have seen the special method `open(...)` before, and again now on lines 3 and 7. What do we use the `open(...)` method for?

b. What do you think the first parameter of `open(...)` represents?

c. We've seen the second parameter, usually as 'w' or 'r' (and once as, 'a'). What might the letters in `open(...)`'s second argument on line 3 mean?

FYI: The special method `open(..)` accepts a variety of characters to specify the open *Mode*. The default mode is 'r'. A 'b' is *Binary Mode* and returns objects as bytes.

- d. If we used a text editor to open 'save.pickle', what might we see?

- e. If we replaced 'save.pickle' on line 3 and 7 to 'age.p', what would the code do differently?

f. The characters below are what you see when you open age.p with emacs (a raw text editor):

```
\200^C}q^@(X^E^@^@^@ dizzyq^AK^FX^E^@^
@^@pixelq^BK^@X^E^@^@^@tallyq^CK^Au.
```

Why is the data stored in age.p not human-interpretable?

Application Questions: Use the Python Interpreter to check your work

- 1. a. Write some code that creates a list object, fav, with all your favorite things about spring. Add code to pickle fav so it can be accessed again, even after the script stops running.
`if __name__ == "__main__":`

- b. Add functionality to your code from 1a. that loads your pickled list. Your code should print an error message if it loads an empty list.

FYI: *Persistent objects* are those objects which survive between successive invocations of a program.

2. How does pickling relate to object persistence?
