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.

Critical Thinking Questions:
1. Examine the code in the programmer’s terminal below.

```
Terminal
0 > python3
1 >>> age = {'dizzy': 6, 'pixel': 0, 'tally': 1}
2 >>> exit()
3 > python3
4 >>> age["dizzy"]
```

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

b. What will happen when line 4 is executed?

c. Write code to insert prior to line 4 to achieve the programmer’s intent from question 1a.

d. If you could invent a feature for python to avoid having to repeat your response in 1c., 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:

```
0 > python3
1 >>> age = {'dizzy': 6, 'pixel': 0, 'tally': 1}
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 6
```

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

________________________________________________________________________

b. What does the ‘6’ on line 9 refer to?

________________________________________

________________________________________

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

______________________________________

____________________________

d. What might be stored in the newage variable?

______________________________________

____________________________

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?

_________________________________________________________________

d. If we replaced “save.pickle” on line 3 and 7 to “age.p”, what would the code do differently?

_________________________________________________________________

e. If we used emacs to open “save.pickle” (now “age.p”), what might we see?

_________________________________________________________________

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.
4. The characters below are what you see when you open age.p with emacs:

\200\C\}q@\(X^E@\@dizzyq\AKFX^E@^@
@\pixelq\BK@X^E@\@tallyq\CKAu.

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

c. If you wanted to try and make the data in age.p human-interpretable, what line of code might you try?

(Note: pickling requires bytes, so it won’t be possible, but what would you write if you wanted to see if it was possible?)

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.

```python
if __name__ == "__main__":

```

1. 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.

```python
```

FYI: Persistent objects are those objects which survive between successive invocations of a program.
2. How does pickling relate to object persistence?

______________________________________________________________________________
______________________________________________________________________________
______________________________________________________________________________
______________________________________________________________________________
______________________________________________________________________________
______________________________________________________________________________