Pickle (Irene)

Python Reference Document for pickle module:

<https://www.pitt.edu/~naraehan/python3/pickling.html>

**What is the pickle module?**

Pickling is the process of serializing Python objects into byte streams (0s and 1s).

We can also unpickle and turn the byte streams back into objects. Pickling supports almost all data types. In other programming languages it is also referred to as serialization, marshalling, or flattening. Pickle only works with Python.

**Why use pickle?**

Pickling can be used to save data to a disk in an application where you need persistency. Can be used to send data over a Transmission Control Protocol (TCP) connection in a multi-core or distributed system, or to store python objects in a database.

**Using the pickle module:**

To use you must import pickle

To pickle:

* import pickle

my\_dictionary = {“band”: "Ford", “model”: "Mustang", “year”: 1964}

pickle\_out = open("example.pickle", "wb")    # (file.name, “writing bytes”)

pickle.dump(my\_dictionary, pickle\_out)  #(what you want to dump, where you want to dump it)

pickle\_out.close()

To unpickle:

* Import pickle

pickle\_in = open("example.pickle", "rb")     # (file.name, “reading bytes”)

example\_dict = pickle.load(pickle\_in)     # load file content as example\_dict

**pickle.dump()**

Pickle.dump is the method for saving data to the designated pickle file.

**pickle.load()**

Pickle.load is the method used to load the designated file.

**Warning:** The pickle module is not secure. Only unpickle data you trust!

Example questions:

1. Complete the code bellow:

>>> import pickle

>>> x = {1: “a”, 2: “b”, 3: “c”}

>>> f = open(\_\_\_\_\_\_\_)

1. (‘test.pickle’ , ‘w’)
2. (‘test.pickle’, ‘rb’)
3. (‘test.pickle’, ‘wb’)
4. (‘test.pickle’, ‘r’)

    2. What does the pickle.dump() method return?

      A. returns a bit object for the pickled Python object

      B. returns a binary serialization Python object

      C. returns a Python object unpickled

                  D. returns a nonbinary serialization Python object