More on Pickling: Serializing for I/O Streams

23 Fri 2009
CMPT140
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Quiz 3 20pts, 15min

- (see quiz3.html)

Serialization

- Python's I/O framework is flexible enough to support I/O with any kind of stream:
  - Files, network sockets, other programs (IPC), hardware drivers (e.g., robotics)
- To send data, it must be serialized: converted into a stream of bytes which we can .write() to a file handle or I/O stream
- Different data types serialize differently
  - If we make our own new data types, we have to specify how to serialize
- Pickle is Python's framework for serialization

How to pickle/unpickle?

- import pickle
- Open the file (read or write mode)
- Write the object: pickle.dump(obj, file)
- Read an object: obj = pickle.load(file)

- Pickled objects can be interspersed with regular text in the file; you just have to seek() to the right spot where the pickled object should be
- Get the pickled object without writing to file:
  - pickledObj = pickle.dumps(obj)
  - This is just a string; you can then write() it

What can be pickled?

- None, True, False
- ints, long ints, floats, complex, strings,
- Tuples, lists, sets, dictionaries of picklable objs
- Functions defined at the top level of a module
- Classes (user-defined types) defined at the top level of a module
- Instances of such classes, when everything in their __dict__ is picklable
  - __dict__ tells you what's in a container obj