Objects Review

9 Nov 2009
CMPT140
Dr. Sean Ho
Trinity Western University

Quiz 04 (10 mins, 20 pts)

- Convert 11001011 from binary to both hexadecimal and octal, in Python form. [5]
- Express 4 mebibits/sec in bytes/sec [4]
  - (express your answer in powers of 2)
- Describe and contrast a library's header (DEF) file with its implementation (IMP) file. What does this look like in Python? [6]
- What does the .flush() method do on file handles? When might it be needed, and why? [5]

Quiz 04: answers #1-2

- Convert 11001011 from binary to both hexadecimal and octal, in Python form. **[5]**
  - Hex: (1100)(1011) = 0xCB
  - Oct: (011)(001)(011) = 0313
- Express 4 mebibits/sec in bytes/sec **[4]**
  - Binary units: 1 mebibit = 220 bits (~1.05Mb)
  - 8 bits = 1 byte
  - 4 mebibits/s = $4*2^{20}$ bits/s = 2^{22} bits/s = 2^{19} bytes/sec

Quiz 04: answers #3

- Describe and contrast a library's header (DEF) file with its implementation (IMP) file.
  What does this look like in Python? **[6]**
  - Header: “user manual” for programmers, declares what functions/classes are provided by the library, and how to use them
  - Implementation: hidden from users of the library; bodies/code of the functions
  - In Python: both header and implementation are in the same *.py file

Quiz 04: answers #4

- What does the .flush() method do on file handles? When might it be needed, and why? [5]
  - Buffered output: .write() does not necessarily commit the changes to the file on hard disk right away, for performance reasons
  - .flush() forces the changes to be committed, waiting for the hard disk to finish writing before proceeding. Useful if you want to be sure the changes have been written (e.g., auto-save if program might crash)

Classes and instances

- We define (declare) object classes (types).
  A class is a user-defined type, containing:
  - Attributes: data stored in each object
  - Methods: operations involving the object
    - Constructor method: how to set up a new object
    - Destructor method: how to destroy an object cleanly
- Then instantiate the class (declare variables)
- e.g.: joe is a variable of type Student
  - joe is the instance; Student is the class

Default params evaluate once

- Functions can have default parameter values:
  - def __init__(self, f='', l=''):
- Default values are evaluated once at declarat'n
  - def __init__(self, f='', l='', bday=Date()):   # wrong!
    - This uses one shared Date object as the default birthday for every student!
- Use None as the default value, and instantiate a new object as the default value at run time:
  - def __init__(self, f='', l='', bday=None):
    - if bday == None:
      - self.birthdate = Date()

Listing all entities in a class

- Special Python attribute '__dict__'
- Dictionary of all entities in the object
  - For module: lists all methods, constants, etc. __module__, __doc__ (docstring)
    - import math
    - math.__dict__
    - Student.__dict__
  - For object: lists all attributes

    joe.__dict__: {'firstName': 'Joe', 'lastName': 'Smith', 'GPA': 3.8}