CSCI 150 – Fall 2006 Handout: HW07
Introduction to Object Oriented Programming
Michael H. Goldwasser
Saint Louis University Monday, 4 December 2006

HW07

Problem A (5 points). Using the new style, as in Figure 9.2(b), draw a diagram of the final configuration which result after executing the following program.

```
v = Account( )
w = Account( )
x = v
v.deposit(100)
w.deposit(200)
x.deposit(400)
v.withdraw(25)
w.withdraw(25)
x.withdraw(25)
```

Problem B (4 points). The following method is designed to accept a list sent by the caller and change the state of that list so as to empty the list (if not already empty).

```
def clear(data):
  """Empties the given list"""
  data = [ ]
```

Unfortunately, this implementation is flawed. Explain in your own words why the given code fails.

Problem C (5 points). The __contains__ method of the list class is used to determine whether a given element is contained within a list. Given our knowledge of a list as a sequence of references, we might re-examine the precise semantics of this built-in behavior. There are two possible interpretations of containment: (A) that it checks whether a given object is itself an element of the list; (B) that it checks whether the list contains an element which is equivalent to the given parameter.

Your job is to perform experiments to conclusively determine which of these interpretations is supported by Python's lists. In explaining your solution, give the text of an interpreter session (including Python's responses) which constitutes your "experiment." Then clearly state your conclusion and why the experiment supports your conclusion.

Problem D (6 points). The traditional phrase "my house is your house," is used to welcome a guest into my home, suggesting that they consider my house as if it were their own posession.

Explain in your own words the analogous real-world semantics expressed by each of the following examples of Python syntax.

(a) myHouse **is** yourHouse

(b) myHouse == yourHouse

(c) myHouse = yourHouse

(d) myHouse = copy(yourHouse)

(e) myHouse = deepcopy(yourHouse)

(f) None of the above syntaxes properly capture the spirit of the original statement, "my house is your house." Give a Python syntax which captures that meaning.

Extra Credit (2 points) On page 380, we gave an implementation of the list.pop method which allows for support of the syntax groceries.pop(5) as well as groceries.pop( ). Here is a very similar implementation, but unfortunately a flawed one.

```
def pop(self, index = None):
  if not index # nothing sent by caller
    index = len(self) − 1 # so use this instead
```

This attempt has a *very* subtle, yet critical bug. Suggest a specific setting in which the outward behavior of this implementation would differ from that of the standard implementation. Explain why the flawed syntax leads to that behavior.