

Name: _____ Andrew Id: _____

15-112 Spring 2021 Quiz 08
25 minutes.

1. Code Tracing [5 pts] Indicate what the following program prints. Put a box around your final answer.

```
class A(object):
    def __init__(self):
        self.b = "Kitten"

    def f(self):
        print("Cat")

    def g(self):
        print("Dog")

    def h(self):
        print(f"{self.b}: Horse")
        self.g()

class B(A):
    def g(self):
        print("Camel")
        super().g()

class C(B):
    def __init__(self, a):
        self.b = a
        self.f()

    def h(self, s):
        super().h()
        print(f"{s}: Lion")

def ct():
    c = C("Goat")
    c.h("Cow")

ct()
```

2. Reasoning Over Code [3 pts] For the following code, find a value `L` which will cause `roc(L)` to return `True`. Put a box around your final answer.

```
class R(object):
    def r(self, L):
        if len(L) == 1:
            return L[0] > 10 and L[0] < 99 and \
                L[0] % 9 == 0 and int(str(L[0])[0]) == 2
        elif len(L) == 2:
            return self.r(L[:len(L)//2])
        else:
            return self.r(L[len(L)//2:])

def roc(L):
    assert(len(L) == 5)
    t = R()
    return t.r(L)
```

3. Free Response [12 pts] Write the classes `Toy` and `Stuffie` so that they pass the following test cases. You may not hardcode any test cases. **For full credit you must use inheritance appropriately as well as avoid duplicating code needlessly.**

```
# A basic toy has an owner
t = Toy("Susy")
assert(t.getOwners() == "Susy")
assert(str(t) == "Toy (owner=Susy)")

# Toys can also have more than one owner
t.addOwner("Johnny")
t.addOwner("Zed")
t.addOwner("Albus")
# The order the owners are listed matters...
assert(t.getOwners() == "Albus,Johnny,Susy,Zed")
assert(str(t) == "Toy (owner=Albus,Johnny,Susy,Zed)")

# Toy properly handles equivalence checking
n = Toy("Johnny")
n.addOwner("Albus")
n.addOwner("Susy")
n.addOwner("Zed")
assert(t == n)
assert(t != Toy("Billy"))
assert(t != "Johnny")

# A basic stuffie has an owner and a name
s = Stuffie("Hamoodie", "MyBear")
assert(str(s) == "Stuffie (name=MyBear, owner=Hamoodie)")

# Stuffies, like Toys, can also have multiple owners
s.addOwner("Fatima")
# Just like Toys, order of owners listed matters...
assert(s.getOwners() == "Fatima,Hamoodie")
assert(str(s) == "Stuffie (name=MyBear, owner=Fatima,Hamoodie)")

# Stuffie properly handles equivalence checking
s = Stuffie("Hamoodie", "MyBear")
assert(s == Stuffie("Hamoodie", "MyBear"))
assert(s != Toy("Hamoodie"))
assert(s != Stuffie("Billy", "MyBear"))
assert(s != Stuffie("Hamoodie", "YourBear"))
assert(s != 42)

# Verify some inheritance rules...
assert(isinstance(t, Toy) == True)
assert(isinstance(t, Stuffie) == False)
assert(isinstance(s, Toy) == True)
assert(isinstance(s, Stuffie) == True)
```


